

(19) 日本国特許庁(JP)

## (12) 公開特許公報(A)

(11) 特許出願公開番号

特開2004-220556

(P2004-220556A)

(43) 公開日 平成16年8月5日(2004.8.5)

(51) Int.Cl.<sup>7</sup>**G06F 9/445**  
**HO4N 1/00**

F I

G06F 9/06 650  
HO4N 1/00 106C  
G06F 9/06 640A

テーマコード(参考)

5B076

5C062

審査請求 未請求 請求項の数 25 O L 外国語出願 (全 113 頁)

(21) 出願番号 特願2003-358032 (P2003-358032)  
 (22) 出願日 平成15年10月17日 (2003.10.17)  
 (31) 優先権主張番号 60/319,623  
 (32) 優先日 平成14年10月17日 (2002.10.17)  
 (33) 優先権主張国 米国(US)  
 (31) 優先権主張番号 60/319,624  
 (32) 優先日 平成14年10月17日 (2002.10.17)  
 (33) 優先権主張国 米国(US)  
 (31) 優先権主張番号 60/319,625  
 (32) 優先日 平成14年10月17日 (2002.10.17)  
 (33) 優先権主張国 米国(US)

(特許庁注:以下のものは登録商標)

1. Linux

(71) 出願人 596170170  
 ゼロックス コーポレイション  
 XEROX CORPORATION  
 アメリカ合衆国 コネチカット州 スタン  
 フォード、ロング・リッジ・ロード 80  
 O  
 (74) 代理人 100075258  
 弁理士 吉田 研二  
 (74) 代理人 100096976  
 弁理士 石田 純  
 (72) 発明者 マイケル アール ファースト  
 アメリカ合衆国 ニューヨーク ロチェス  
 ター ダートマウス ストリート 268

最終頁に続く

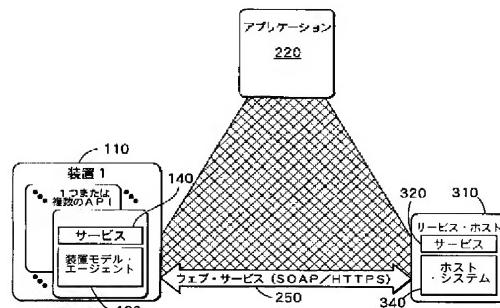
(54) 【発明の名称】装置モデル・エージェントとアドオン構成要素を含む、分散加入サービス、自動化供給物保守、および装置独立のサービスの実施を可能にする方法および装置、装置側の低コスト組み込みプラ

(57) 【要約】 (修正有)

【課題】ユーザに納入後の、装置の改善をフレキシブルに供給する方法。

【解決手段】分散型システム1は、記録装置110などが、装置中心サービス140に加入しそのサービスを動作させることを可能にする。装置モデル・エージェント120は、装置110が、サービス提供者のサービス・ホスト310と対話して、供給品保守、ユーザ・ヘルプ、およびサービス加入を自動化することを可能にする。装置モデル・エージェント120は、装置110に組み込むあるいは装置110に接続されたアドオン構成要素に配置、個別の装置によってプロキシ220として実行されてもよい。装置モデル・エージェント120は、装置110が利用可能なサービス140の実行時環境を、共通インターフェースを介して共通構造と共に提供する。

【選択図】図5



**【特許請求の範囲】****【請求項 1】**

サービス・ホストから少なくとも 1 つの使用可能なサービスを提供することができ、少なくとも 1 つの装置固有の提供者アプリケーション・プログラム・インターフェースを含み、かつ、装置固有のステータス情報を有する少なくとも 1 つの装置と、  
少なくとも 1 つのサービス層と、

少なくとも 1 つのサービスが実際に動作する少なくとも 1 つのサービス環境と、少なくとも 1 つの共通情報管理アプリケーション・プログラム・インターフェースと、少なくとも 1 つの装置モデル・エージェントと、少なくとも 1 つの共通提供者・アプリケーション・プログラム・インターフェースとを含む装置独立ランタイム環境と、  
を含む分散システム・アーキテクチャ。

**【請求項 2】**

請求項 1 記載の分散システム・アーキテクチャであって、  
少なくとも 1 つのサービス層が、少なくとも 1 つの装置に利用可能にされた少なくとも 1 つのサービスを含む分散システム・アーキテクチャ。

**【請求項 3】**

請求項 1 記載の分散システム・アーキテクチャであって、  
少なくとも 1 つの装置独立ランタイム環境が、記録装置内に配置されており、記録装置が、少なくとも 1 つの装置のうちの 1 つである分散システム・アーキテクチャ。

**【請求項 4】**

請求項 1 記載の分散システム・アーキテクチャであって、  
少なくとも 1 つの装置独立ランタイム環境が、記録装置に接続されたサーバ内に配置され、記録装置が、少なくとも 1 つの装置のうちの 1 つである分散システム・アーキテクチャ。

**【請求項 5】**

請求項 4 記載の分散システム・アーキテクチャであって、  
サーバが、主な機能が装置独立ランタイム環境と関連付けられていないが、装置独立ランタイム環境をホストするアプリケーションをホストする分散システム・アーキテクチャ。

**【請求項 6】**

請求項 1 記載の分散システム・アーキテクチャであって、  
少なくとも 1 つのサービス環境とサービス層が、記録装置に接続されたサーバ上にあり、記録装置が、少なくとも 1 つの装置のうちの 1 つであり、サーバが、主な機能がサービス層および環境と関連付けられていないがそのサービス層および環境をホストするアプリケーションをホストする分散システム・アーキテクチャ。

**【請求項 7】**

請求項 1 記載の分散システム・アーキテクチャであって、  
装置モデル・エージェント、少なくとも 1 つのサービス環境、およびサービス層が、記録装置に接続されたサーバ上にあり、記録装置が、少なくとも 1 つの装置のうちの 1 つであり、サーバが、主な機能が装置モデル・エージェント、サービス層およびサービス環境と関連付けられていないが、その装置モデル・エージェント、サービス層およびサービス環境をホストするアプリケーションをホストする分散システム・アーキテクチャ。

**【請求項 8】**

装置独立サービスを提供する方法が、  
共通の装置インターフェースを提供する段階と、  
共通の情報モデルを提供する段階と、  
共通の装置インターフェースおよび情報モデルを使用して、装置内のサービスを統合する段階と、  
装置固有の違いを共通の装置インターフェースの後ろに隠す段階と、  
を含む方法。

10

20

30

40

50

**【請求項 9】**

請求項 8 記載の方法であって、

共通の装置インターフェースを提供する段階が、それぞれのサービスの所定の拡張機能を備えた分散モデル・タスクフォース共通情報モデルを使用する段階を含む方法。

**【請求項 10】**

請求項 9 記載の方法であって、

共通情報モデルを提供する段階が、装置とそれぞれのサービス間の互換性を高める所定の拡張機能を備えた分散モデル・タスクフォース共通情報モデルに基づく段階を含む方法。

**【請求項 11】**

プラットフォームを提供する方法であって、

アクセス・モジュールを提供する段階と、

サービスが、アクセス・モジュールによって装置の組み込まれた計算能力、データ、および機能を使用できるようにする段階と、

アクセス・モジュールを共通の方式で配置する段階と、

を含む方法。

**【請求項 12】**

請求項 11 に記載の方法であって

新しく配置されたサービスを、ホスティング・プラットフォーム用のソフトウェア・リリースと非同期して受け入れる段階をさらに含む方法。

**【請求項 13】**

請求項 11 に記載の方法であって、

サービス・プラットフォームをホスト・プラットフォームに組み込む段階をさらに含む方法。

**【請求項 14】**

請求項 11 に記載の方法であって、

アドオン構成要素内のサービス・プラットフォームをホスト装置に配置する段階をさらに含む方法。

**【請求項 15】**

請求項 14 に記載の方法であって、

少なくとも 2 つのインターフェースを介してアドオン構成要素をホスト装置に接続する段階をさらに含む方法。

**【請求項 16】**

請求項 14 に記載の方法であって、

アドオン構成要素をネットワークに接続し、それによりホスト装置に装置サービスに関する機能を提供する段階をさらに含む方法。

**【請求項 17】**

請求項 14 に記載の方法であって、

アドオン構成要素を介してホスト装置のすべてのネットワーク接続を提供する段階をさらに含む方法。

**【請求項 18】**

請求項 11 に記載の方法であって、

ユーザ環境に配置された少なくとも 1 つのアプリケーションを、少なくとも 1 つの装置とサービス・ホストの間のサービス・プロキシとして使用する段階をさらに含む方法。

**【請求項 19】**

請求項 18 に記載の方法であって、

使用する段階が、少なくとも装置からサービス・プロキシにデータを第 1 のプロトコルで送り、サービス・プロキシからサービス・ホストにデータを第 2 のプロトコルで送る段階を含む方法。

**【請求項 20】**

10

20

30

40

50

請求項 1 9 に記載の方法であって、  
第 1 のプロトコルが、S N M P である方法。

【請求項 2 1】

請求項 1 9 に記載の方法であって、  
第 1 のプロトコルが、無線通信プロトコルである方法。

【請求項 2 2】

請求項 1 9 に記載の方法であって、  
装置ベースのサービスの対応システムを使用して、サービス・ホストにおける装置変動を管理する段階をさらに含む方法。

【請求項 2 3】

請求項 1 9 に記載の方法であって、  
ユーザ環境にあるサーバにサービス管理を統合する段階をさらに含む方法。

【請求項 2 4】

請求項 1 9 に記載の方法であって、  
ユーザがサービスを管理することができる U I を提供する段階をさらに含む方法。

【請求項 2 5】

サービスが、装置にアクセスしている間に機能を提供するように意図された装置と実質的に独立にサービスが動作できる環境を提供する装置モデル・エージェントであって、少なくとも 1 つのサービス・ホストと通信して、自動化された供給物の保守とサービスの加入および配置を可能にする装置モデル・エージェント。

【発明の詳細な説明】

【技術分野】

【0 0 0 1】

実施形態は、電子式複写記録装置 (marking device) 、ファクシミリ装置、走査装置、多機能装置などに関する。詳細には、実施形態は、係る装置がユーザに提供することができるサービスの実施および配布に関する。

【背景技術】

【0 0 0 2】

本出願は、2002年10月16日に出願された米国暫定特許出願番号 60/319,622、ならびに2002年10月17日に出願された 60/319,623、60/319,624、60/319,624、60/319,625に対して優先権を主張する。

【0 0 0 3】

記録装置や他の業務装置の設置は、その装置のライフサイクルの大部分において第 1 歩に過ぎない。ほとんどの装置は、製品所有者（ユーザ）、製品の製造業者、および／またはサードパーティ供給業者の間の継続的な業務プロセスに関わらされる。記録装置を製造する会社は、一般に、ユーザの文書をサポートする製品とサービスを有し、ユーザが、ずっとその提供物を使用しその提供物に満足し続けることを望む。この販売後の期間は、製造業者とユーザの間に互いに利益となる長期間に渡る強い関係を築く機会を提供する。この販売後の関係は、装置がユーザのために何をするかというだけでなく、装置がそれをどのように行い、製造業者が装置をどのようにサポートし、製造業者がユーザをどのように扱い、装置を所有し使用することが全体としていかに容易かということにより定義することができる。これを理解することにより、実施形態は、ユーザが使用している装置をサポートするサービスを受けるためにユーザの補足的なニーズに応え、すなわち、様々な実施形態において、販売後のライフサイクル、破損／修復のニーズ、および統合的な業務プロセスが対処される。そのようなプロセスは、破損／修復サービス（修理）から、消耗品および供給品の継続的な供給、製品アップグレード、拡張、ならびにソリューションや他の提供物への統合まである。従来、このような販売後のプロセスは、本質的に手作業であり、装置の所有者／ユーザに、必要なときに製造業者とサプライヤに限られた情報を中継する際に積極的な役割を演じることを要求していた。

10

20

30

40

50

**【0004】**

多くの電子式複写記録装置、ファクシミリ装置、走査装置、多機能装置などは、ユーザが使い方または回避の仕方を覚えなければならないプロセスを支援するサービスを提供する。また、装置によっては、ユーザをいらいらさせるメータ読み取りや他の種類の保守を必要とするものがある。メータ読み取りの場合、ユーザは、メータを毎月読み取り、その結果を、例えばファックスや電話で供給業者に伝えなければならない場合がある。さらに、ユーザは、用紙、トナー、他の材料の供給品を手作業で確認し、新しい材料を注文しなければならない。装置によって提供されるサービスの数は、ユーザを困惑させ、それにより、ユーザは、装置が複雑すぎて覚えることができないと考えることがある。さらに、ダウンタイムやその他の不都合を回避するために、ユーザは、修理の電話をし、サービスの到着を待って自分の装置を修理するよりも自分自身で小さな修理を行いたいことがある。

**【0005】**

モデムの出現によって、ユーザまたはユーザ・サイトにあるハイエンド製品が、この対話モデルを変更する電話線によって製造業者に接続された。広く普及したインターネット接続の到来とネットワーク接続製品の急増により、装置を販売後の業務プロセスと統合するフレキシブルかつ強力な手法を提供する新しい機会が現れた。ネットワーク接続は、電話線接続の欠点のうちのいくつかを解決しているが、今まで示されたシステムには、そのような初期のシステムのために開発された対話モデルと関連した多数の制限がまだある。

**【発明の開示】**

10

**【発明が解決しようとする課題】**

20

**【0006】**

30

現行のシステムの欠点は、通信方法とシステム・アーキテクチャとの結びつきが厳格であること、配置と統合戦略が画一的であること、また既に配置済みの装置に一般にサポートがないことがある。既に配置された装置のサポートを提供するシステムは、一般に、既に配置済みの装置と新しい装置との扱い方の間に一貫性がない。さらに、このシステムは、一般に、機能、プロセスおよびワークフローの迅速なアップグレード、拡張、カスタマイズおよび進化の機能を含んでおらず、基本的な業務プロセスに制限されることが多く、外部サービスとソリューションAPIを安定的に提供することができない。一般に、またほとんど全体的に、システムは、装置を、使用可能にされたサービスへの積極的な関与物ではなく単純な情報レポートリレポートリとして扱う。装置は、競争力を維持するためにその主要な機能セットを強化させ続けなければならない。例えば、ドキュメント・システムでは、装置の競争力を高めるために強化される特性は、一般に、速度、供給速度、画像品質、および文書ワークフローである。しかしながら、装置とユーザと供給業者間の販売後の対話が増えており、製品をソリューションとサービスに組み込みまたその逆方向に組み込む能力が、市場において装置を差別化するポイントになってきている。近い将来、装置の成功と価値は、販売後ライフサイクルに積極的に関与する装置の能力、ソリューション提供物とシームレスに一体化する能力、およびユーザのニーズと要件に基づいてカスタマイズし拡張する能力で計られるようになる可能性が高い。そのような装置の能力の成果は改善され、ユーザに使い易くなり、製造業者によるサポートの有効性が高まり、ユーザの全体的な満足度が高まる。

40

**【0007】**

ここ数年間の業界の一般的傾向は、リモート・サービスを提供してユーザの満足度を高めかつ操作費用を削減することによって、記録装置に組み込まれた高い計算性能と高い接続性を利用することであった。接続されたインテリジェント製品のこの傾向は、サーバや他のミッション・クリティカルな情報技術(IT)関連のハードウェア上のリモート・サービスの実現から始まり、記録装置を含む様々な他の業界に広まってきた。そのようなリモート・サービスは、製造業者とユーザの両方にとって価値ある企画を提供する。適切に実現されたとき、そのようなサービスは、製造業者にとって大きなコスト削減を可能にし、さらにユーザにとって販売後より良い経験を可能にする。

50

**【0008】**

この移行は、同時に起こるいくつかの要因およびニーズによって促進される。競争圧力と改善された内部業務プロセスのニーズによって、現地にある製品との新しい対話方法が必要になり、またサービスとサポートの責任を持つ場所の移動が必要になる。同様に、製造業者とユーザは、新しい機能／サービスを構成し追加して、目前にある問題を迅速に解決し新しい機能を迅速に配置できることを好む。このプロセスの単純化と高速化によって、寿命が延び、配置された装置の価値が高まり、ユーザの満足と生産性が維持される。製造業者は、そのような機能を新しい装置と既に配置済みの装置に提供できなければならぬが、製造業者は、全ての種類において最高になることはできず、装置は、サードパーティの部品または競合する部品を容易に組み込むことができなければならない。1つのサイズがすべてに適合するとは限らず、個々のユーザ・ニーズに適したソリューションを構成する能力を製造業者に提供する複数の配置構成が必要である。また、製造業者は、ソリューションの挙動を、管理とサポートが可能なように複数の構成にわたって一貫させることができなければならず、それにより、ユーザは主導権を維持する。

**【0009】**

そのようなタイプのサービスに関するユーザの好みとニーズの決定に焦点を合わせた研究は、最終的に、ユーザが、記録装置、請求書作成システムおよび供給チェーンに満足し続ける方法を強化する提供物の新しい機能のニーズを指摘している。また、この研究は、ユーザが、そのようなサービスを強く望み、セキュリティのハードルを越えてその機能を実現するために製造業者と一緒に作業する意志があることを示している。詳細には、研究から、ユーザのほぼ3分の1は、そのなりリモート・サービスが、次の購入時にユーザを所定の装置ブランドに忠実にさせる可能性が高いことが分かった。ほとんどのユーザは、自分の装置のリモート・サービス機能を得るためにお金を支払う意志があり、またユーザが共有データによって何らかのレベルの制御を有し、指導された自己修復、ソフトウェアの自動ダウンロード、およびリモート供給品／サービス解析または予測に特定の関心を示している限り、インターネットを介してサービス提供者にデータを送ることによりきわめてまたはある程度満足する。

**【0010】**

さらに、リモート・ソリューションの最新技術の解析により、記録装置の製造およびリモート・ソリューションの市場にいる主な当事者はすべて、ある程度のリモート・サービス機能を提供し、その機能の拡張を重要視してきていることが分かった。オフセット印刷市場では、印刷機と周辺装置にリモート・サービスを組み込むことは、事業コストと見なされている。

**【0011】**

現在のシステムより前にユーザに提供されたサービスは、特定の製品系列の範囲内で端末同士（end-to-end：エンド・ツー・エンド）で組み立てられ管理されていた。これには、製品自体だけでなく、ジョブをやり終えるために必要なインフラストラクチャ、サービス、およびバックオフィス接続の開発に投資する製品チームが必要であった。この作業は、長期間維持するのがきわめて難しく、また製品系列のなかで重複することが多かった。

**【0012】**

例えば記録装置などの装置とユーザの関係を単純化することによって、ユーザの経験を大幅に強化することができる。実施形態は、現在手作業で行われかつ／または不均一な業務プロセスを自動化し、またユーザの要求に対応するための新しいワークフローを提供することができる。これは、例えば、装置が、ユーザの主導権を維持しながら、その装置のライフサイクルと付加価値サービスに積極的に関与できるようにする実施形態を使用することによって達成される。実施形態は、これを、分散管理タスクフォース（Distributed Management Task Force）や共通情報モデル（Common Information Model）（CIM）に基づく規格といった規格アーキテクチャを使って行い、実施形態を使用しかつ／または実施形態と互換性のあるすべての装置に関してサービスを一度書き込むことを可能にし、製品ごとに新しいサービスをモジュール式に容易に追加できるようにする。

10

20

30

40

50

**【課題を解決するための手段】****【0013】**

これらの目的を達成するために、実施形態は、共通サービス・モデル、複数の異種装置により動作するサービス、ならびに物理的構成、論理的構成および動作的構成におけるフレキシビリティを提供する。装置は、ユーザに強化された販売後の経験を提供する際に積極的な役割を持つ。実施形態は、ユーザと製造業者の両方のバックオフィス・プロセスへのシームレスな統合を得ることができる。

**【0014】**

より詳細には、実施形態は、装置をソリューション提供物に接続するためのフレキシブルなエンド・ツー・エンド・システムを含む。装置の変更をバックオフィスの変更から分離しながら、様々な物理的位置および構成における多数の配置の選択肢が、現地の装置と新しい製品の両方に関して最も広い装置有効範囲と迅速な機能配置を可能にすることができる。

**【0015】**

実施形態のシステムは、互換性のあるすべてのプラットフォームに再使用することができ、個々のプラットフォームが、すべてのバックオフィス・システムを再構成する必要がなくなる。各プラットフォーム・チームは、例えば実施形態のDMAを組み込みかつ／または特定のサービス・トランザクション・プロトコルに従うことによって、前に述べ実施形態によって意図されている方法のうちの1つによって製品を使用可能にするだけよい。

**【0016】**

装置、アドオン・モジュール、および装置プロキシに組み込まれたエージェント・ソフトウェア構成要素は、共通装置モデル、共通情報管理（CIM）アプリケーション・プログラミング・インターフェース（API）、および装置サービスが動作させることができる環境を提供する。通信機構の共通の抽象化より、システムを物理トランスポート結合ノードから独立させることができる。サービス・モデルは、装置の近くで動作するサービスとそのライフサイクルをサポートし、サービスとソリューションを有効に管理しカスタマイズする方法およびプロセスを含む。その結果、エージェントに一回作成されたサービスは、エージェントを含む任意の装置、アドオン・モジュールまたはプロキシ上で動作することができる。これにより、装置および装置プロキシを配置し、サービスの観点から共にシームレスに動作できるようにするシステムと、ユーザと供給業者の両方から入力がある装置に基づいたサービスのポリシ（policy）に基づいた予備との対応が得られる。組み込みサービス・エージェントは、ソリューション提供物において積極的な役割を持ち、必要な機能を提供するために分散ソリューションおよび／またはネットワーク・アクセス可能なサーバと共に動作する。サーバは、システムを横断しなければならないメッセージのクリアリング・ハウスを提供し、分散サービスを複数の細分レベルで接続しカスタマイズするのに必要な管理機能を提供する。

**【0017】**

実施形態は、高いユーザの満足と忠誠の他に、経済的な利益を生み出すことができる。実施形態は、高いユーザ自助機能、リモート診断機能および予測機能によるサービス技術者の使用量を少なくすることにより、コスト削減を実現することができる。自動メータ読み取りを含む実施形態において、少ない収集プロセス・インフラストラクチャ、より適切な契約の実施、および誤りに対する少ない備えによって、さらなるコストの削減を実現することができる。さらに、自動化供給物の注文に関する実施形態によって、ある程度は、よりタイムリーで正確かつ妥当な測定によるユーザ・サイトにおける消耗品を追跡する高い精度によって、在庫を減少させることができる。電話注文や、やり取りの数が少なくなり電話時間が少なくなるという観点から、さらなるコスト削減を実現される。最後に、きわめて多くのユーザが、実施形態によって提供されるサービスの料金を喜んで支払うことになるため、実施形態は、新しいサービスからの収益の増加に役立つ。

**【0018】**

10

20

30

40

50

実施形態は、例えば、新しい種類のリモート・サービスを含めることによって、ユーザのニーズと関心に応える。そのようなサービスは、ユーザ環境における装置の高い接続性を利用し、装置自体に組み込まれた計算機能を利用して、装置がユーザ作業プロセスの単純化に関与するようにする。プラットフォームは、共通バックオフィスの統合および作業プロセスをすべて使用するリモート・サービス提供物をモジュールでクロスプラットフォーム式に実現するために使用することができる規格ベースのソリューションを使用可能にする。実施形態において提供することができるサービスのタイプの特定の例には、自動メータ読み取り、自動供給品注文、生産性報告、ソフトウェア・ダウンロード、ユーザ支援自助機能、リモート診断、および予測がある。

#### 【0019】

実施形態は、装置（プリンタ、スキャナ、レポジトリ、さらに他のサービスおよびソリューション）と、それらの装置を所有し、使用し、サポートし、購入し、アップグレードし易くするライフサイクルのサポートに存在するサービス・クラスを含む。市場調査により、このようなサービスが、ユーザに対する装置の価値を高め、また製品の寿命の間ずっとユーザの満足度と高める可能性があることが分かった。その結果として、このサービスは、新しく購入するときに、ユーザからのユーザの忠実性と配慮を大きくするはずである。

#### 【0020】

そのようなサービスは、実施形態において、組み込まれた装置インテリジェンスを含む新しい装置機能を利用し、多数のネットワーク接続された個体群を利用し、情報技術の進歩を調査して、装置が、販売後のライフサイクルでより積極的な役割を演じることができるようにして、自動化され拡張された1組の機能を使用可能にする。

#### 【0021】

実施形態は、供給業者がユーザにそのようなタイプの販売後サービスを有効かつ効率的に提供できるようにする基本的な1組の構成要素とその相互接続を提供する。プラットフォームのために定義された高い目標は、初期の構成要素およびサービスのアーキテクチャおよび開発を促進するために使用してきた。それぞれの詳しい属性が、プラットフォームの大きな4つの目標をサポートする。このシステムの主な構成要素はすべて、提供されるサービスをユーザにとってシームレスに挙動させるために裏側で一緒に動作する。

#### 【0022】

実施形態は、電話、ファックスまたはコンピュータ・ネットワークを介してメータ読み取りの自動報告機能を提供する。さらに、実施形態は、自動的に供給品を監視し供給品が少なくなったときにユーザに警告し、次およびその後の類似の状況での供給品の自動注文を可能にする。さらに、装置が提供するサービスは、ユーザの特定のニーズに合わせることができるが、実施形態によって提供される自動化されたサービス加入、ダウンロード、およびインストールによって、後でユーザの必要に応じて増減することができる。さらに、実施形態は、ユーザに、小さな修理とユーザが交換可能なユニットの交換を含むユーザが実行したい任意の作業を順番にさせることができる。実施形態のさらに他の利点は、中央アプリケーションからの多数の装置の長所を管理する能力である。

#### 【発明を実施するための最良の形態】

#### 【0023】

本発明を概略的に理解するために、図面を参照する。図面において、類似の参照数字は、全体にわたって、同一の要素を示すために使用している。

#### 【0024】

実施形態は、構成要素の物理的および論理的なシステム設計の柔軟性と責任を保証するいくつかのタイプの分散されたソフトウェア構成要素とハードウェア構成要素からなるシステム1を提供する。実施形態は、例えば、ユーザ／ユーザ環境100にある装置110、ユーザのネットワークまたは環境100内にあることができる資産管理システム200、および装置が申し込むことができるサービス320を提供するサービス・ホスト310を含むアーキテクチャを使用する。システム管理およびサービスがシステムに提供され、

10

20

30

40

50

そこで、装置は、その装置自体のサービスとライフサイクルの両方のニーズならびにそのようなサービスとライフサイクルの一部分だけに積極的に関与する。

#### 【0025】

基礎として装置モデル・タスクフォース (Device Model Task Force) (DMTF) 共通情報モデル (Common Information Model) (CIM) を使用して、サービス管理が、共通装置モデル・エージェント (Common Device Model Agent) (CDMA) 120 の積極的な挙動 (active behavior) に追加される。詳細は、例えば図1と図10を参照されたい。各装置110は、例えばDMTF CIMなどの共通中間言語を使用して、装置110のステータスおよび構成 (111の部分)、提供されるサービス140、および他の情報 (例えば、111の追加部分) をサービス・ホスト310に伝えるCDMA120によって、サービス・ホスト310に提示されることが好ましい。DMAは、また、サービス140が常駐する装置上に、サービス140の実行時環境であるサービス環境124を提供し、サービス・ホスト310から提供されるサービスの装置独立性を提供する。したがって、特定のサービス140を、様々な装置110上で1度書き込み実行することができる。これにより、必要に応じて、1対多数構成を使用可能にし、要求し、利用することができる。例えば、実施形態は、後でさらに詳しく述べるように、挙動だけを含む単純なプロキシを含む装置プロキシと、代理の装置モデルとを含むことができる。また、実施形態は、サービス・ホストと直接通信する装置を含むことができる。

#### 【0026】

CDMA120は、サービス環境124、CIM API123、コアDMA122内のCIMOM125とサービス・マネージャ126、および共通提供者 (common provider) API1121を含む。共通提供者API1121は、装置110の装置固有の提供者API1112と通信し、例えばその構成、ステータスおよび供給品レベルを含む装置に関する情報111を取り出す。次に、共通提供者API1121は、そのような情報を、コアDMA122内のCIMOM125とサービス・マネージャ126と、サービス環境124内で実行しているサービス140に利用可能にする。したがって、実施形態において、DMA120は、サービス層141 (サービス140を実行する集合体) と装置固有API1112の間に存在することができ、サービス・ホスト310と直接通信することができる。そのような場合は、サービス・ホスト310と通信できるようにするために、DMA120を、電話線やコンピュータ・ネットワークなどの通信媒体に接続しなければならない。

#### 【0027】

部分的にはDMA120を使用する結果により、実施形態のシステム1は、実質的な通信媒体独立性を達成する。装置110は、ネットワーク、地上電話線、セルラ通信、パケット・ラジオ、ページヤ方式、Blue tooth (商標)、IEEE802.11、または他の適切な通信方式により、サービス・ホスト310および／またはサービス・プロキシと通信することができる。そのような通信は、装置から行われてもホストから行われてもよく、監視されかつ／または監査されてもよく、ユーザの好み、サービス提供物 (service offerings)、および全体的なサービス品質を使用して、特定のシナリオにどの選択が適切かを決定することができる。サービス140は、装置構成から独立することができ、バックオフィスまたは供給業者300が、サービス内容の詳細を提供することができ、サービスの申し込みは、バックオフィス業務プロセスによって発行され有効にされ、新しいサービスの迅速な導入が可能になる。

#### 【0028】

従って、実施形態は、モジュール性、分散計算、サービス・モデルおよびトランザクションの独特的な組み合わせを使用して多くの構成要素から組み立てられたエンド・ツー・エンド・システム1を含む。実施形態は、迅速かつフレキシブルなサービス配置をサポートするシステム・アーキテクチャ全体をモジュール式に使用する。このアーキテクチャは、システム要素、共通インターフェース、およびそれらの間の通信のためのメッセージング・モデルの機能と識別の抽象化を使用するため好都合である。装置サービスには、一貫しか

10

20

30

40

50

つフレキシブルな形で一緒に動作する主たる4つのエンティティがあり、それらのエンティティは、装置自体、ユーザの環境における管理およびサービス・アプリケーション、サービス管理およびサービス構成に特化するバックオフィス・サーバ、およびそのようなサービスを囲む業務プロセス統合サーバおよび業務プロセスである。

#### 【0029】

図10は、実施形態によって提示されたプラットフォームを構成することができる主な構成要素の概略図である。小フットプリント組み込みプラットフォーム140ならびにD MA120に結合されたインテリジェント・エージェント122を配置し、装置110と一体化することができる。装置110のインテリジェント・プロキシは、グループ管理とサービスへの関与を可能にし、独立したアプリケーションまたは他のアプリケーションの一部分としてプラットフォームに含まれてもよい。さらに、そのようなインテリジェント・プロキシは、実施形態の様相と結びつけるためにまだ使用可能にされていないレガシ装置を使用可能にすることができる。インターネットや供給業者インターネットなど、現地の装置および製品を供給業者とその内部システムに確実かつ頑強に接続することができるコンピュータ・ネットワークや他の通信ネットワークに接続された媒体に依存しない通信およびアプリケーション・インフラストラクチャが使用される。さらに、実施形態は、そのようなサービスを可能にするために付加価値のある拡張機能と一体化された1組の業界標準のウェブ・サービス技術を含む。したがって、実施形態は、装置をサポートするエンド・ツー・エンド・システムで実行する1組のサービスを提供する。

#### 【0030】

装置110は、一般に、ユーザ・サイト100に物理的に配置されるが、実施形態は、他のサイトにある装置110にも対応することができ、世界中に分散させてもよい。ローエンド製品からハイエンド・システムにわたる様々な装置110を含むことができる。実施形態は、このシステムでは、例えば3つの主な成功因子(enabler)を提供する装置110を使用する。装置110は、装置固有の違いをインターフェースの後に隠すことによって、サービスとのより容易な統合を可能にするために、共通装置インターフェース(Comm on Device Interface)(CDI)と共通情報モデル(Common Information Model)(CIM)123を提供する。CDIは、CDMA120内で、共通提供者API121として実現することができる。これにより、サービスの再使用が可能になり、システム1の複雑さが大幅に減少する。CDIは、サービスおよびサービス管理のために、サービス提供者および/または製造業者に固有の拡張機能により、分散管理タスクフォース(DMTF)CIMによって指定される。また、CIM123は、診断拡張機能を含む、サービス提供者および/または製造業者が提供することができるサービスを備えた装置に適した特定の拡張機能によってDMTF CIMに準拠することができる。

#### 【0031】

創意に富むシステム1の実施形態における装置110は、例えば、サービス140を装置110に送って装置110の近くで実行することができる組み込みプラットフォーム124を提供する。また、組み込みサービス・プラットフォーム140は、サービス・マネージャ126を介したローカルなサービス管理と、新たに配置されたサービス140をホスティング・プラットフォームのソフトウェア・リリースと同期せずに受け入れる能力とを提供する。これにより、システムの複雑さが減少し、新しいサービスまたは精巧なサービスの現地への配置が高速化される。実施形態において、DMA120が、組み込みプラットフォーム124を提供するが、他のシステムがプラットフォームを提供することもできることが好ましい。DMAは、共通に配置されたサービスにおいて、装置の組み込み計算能力、データ、および機能にアクセスし使用できるようにするソフトウェア・モジュールであることが好ましい。

#### 【0032】

これらの構成要素は、一般に、供給業者300だけでなくユーザの環境100全体にわたって分散される。同時に、これらの構成要素は、製品(装置110やサービス140など)を販売後ソリューション提供物(附加サービス140)に接続するためのフレキシブル

10

20

30

40

50

ルなエンド・ツー・エンド・システム 1 を提供する。システム 1 は、実施形態において、物理的な様々な位置および構成で一連の配置オプションをサポートするアーキテクチャを提供するように設計されている。実施形態は、装置 110 での変化をバックオフィス 300 での変化から分離するように、現地の装置と新しい製品に最も広い装置有効範囲と機能の最も迅速な配置を提供することが好ましい。実施形態は、さらに、装置 110 に組み込まれた独特な付加価値のあるエージェント・ソフトウェア構成要素 DMA 120 、アドオン・モジュール 115 、および／または共通装置モデル 122 と DMTF CIM API 123 と新しい装置サービス環境 124 を提供する装置プロキシ 210 を提供する。さらに、実施形態は、ノードを結ぶ任意の物理トランスポорт（デバイスと供給業者システムの間）からシステムを独立させることができると通信機構の共通の抽象化を実現することができ、これにより、ユーザ要件に基づいてより高いフレキシビリティと配置カスタマイズ性が実現される。実施形態のサービス・モデルは、サービスとソリューションを効果的に管理しカスタマイズする方法とプロセスを含む、「装置の近くで」実行するサービスとその装置のライフサイクルをサポートする。DMA 120 に関して前に述べた実施形態におけるサービスは、そのような任意の使用可能にされた装置 110 またはプロキシ 220 の上で動作することができ、サービスの観点から装置および装置プロキシを相互にシームレスに配置し動作させることができる。実施形態での対応は、ユーザと供給業者の両方が提供した情報に基づいて、装置ベースのサービスのポリシで達成することができ、サービスを迅速に可能にすることができる。

10

20

30

40

#### 【0033】

実施形態の DMA 120 は、ソリューション提供物において積極的な役割をもち、分散ソリューションと共同で動作する。そのような分散装置エージェント 120 は、インターネットや電話システムなどのネットワークを介してアクセス可能な供給業者 300 にあるサーバ 310 と共に作動する。サーバの役割は、ソリューションを横断しなければならないメッセージのクリアリング・ハウスを提供し、分散サービスを複数の細分レベルで接続しカスタマイズするのに必要な管理機能を提供することである。

#### 【0034】

この機能を備えていない既に配置済みの装置 110 の場合、物理システム構成要素 115 を装置 110 に内部的または外部的に追加するオプションが、この機能を使用可能にし、実施形態によって提供される。創意に富むシステム 1 に対して、この方式で使用可能にされた装置 110 は、アドオン構成要素 115 が装置 110 に対して優れたインターフェースを有する限り、機能が組み込まれた装置 110 と違いがないように見える。例えば、そのようなアドオン構成要素 115 を含む実施形態は、記録装置の入出力端子（IOT）に取り付けられ、EPSV 、 PWS 、場合によっては CAN バス・インターフェースを介して IOT に接続され、ネットワークに接続された構成要素を備えることができる。この構成は、 IOT に、装置サービス 140 に関与する機能を与える。次に、 DMA とサービス・プラットフォームに全域にわたる装置機能を提供するために、非標準またはネットワーク・アクセス不可能な API にアクセスしなければならないので、このアドオン構成要素 115 は、装置との 1 対 1 の写像において見つけることができる。

#### 【0035】

入力出力端子（IOT）が別の製造業者によるものである可能性があり、ユーザが多数のデジタル・フロント・エンド（DFE）から選択できるという事実に対応するために、実施形態は、アドオン・システム構成要素を特徴としている。このアドオン構成要素は、例えば、 Linux や Darwin などのオペレーティング・システムや、例えば DMA サービス・プラットフォームを実行する Insignia の Jeode （登録商標） Embedded Virtual Machine などの Java （登録商標）仮想マシンを実行する低コストの組み込みパーソナル・コンピュータ・プラットフォームである。このアドオン構成要素は、 EPSV や PWS などのインターフェースを介して IOT に直接接続することができる。次に、装置は、ユーザの内部ネットワークに接続される。

#### 【0036】

50

ユーザ環境のバックオフィス・サーバにおける装置管理およびサービス・アプリケーション220は、サービス管理およびサービス構成業務プロセス統合サーバならびにそのようなサービスの周辺の業務プロセスに特化する。そのようなシステム・レベルの構成要素とその相互接続の概略図を、図1に示す。

#### 【0037】

実施形態は、具体的には、エンド・ツー・エンド・システム管理および販売後アプリケーション・アーキテクチャにおいてユーザにサービスを提供する際の装置の役割に関する。本発明は主に、装置中心サービス(Device Centric Service) (DCS) プラットフォームにおける装置側テクノロジ・モジュールである装置モデル・エージェント(Device Model Agent) (DMA) 120に焦点を当てている。DMAは、シン(thin)で効率的なアプリケーション/サービス実行環境である。DMAは、そのプラットフォーム自体を修正することなくオンライン・サービス(e-services)をシステム内で設計し、追加し、管理することができるフレキシブルで、拡張可能で、動的なサービス管理システムを提供する。本発明は、特に、DMAからドキュメント・システム装置への利点の統合について示しているが、この概念は、他の分野にも等しく適用可能である。DMA実行時環境は、Java実行時環境と組み込みウェブ・サーバ間のドキュメント・システム装置上にあるシン・ソフトウェア・インターフェース層である。

#### 【0038】

本明細書で述べている装置モデル・エージェントは、ドキュメント・システム装置に以下のようないくつかの機能を追加する。これらの機能の独特な組み合わせによって、システム管理アプリケーションの開発、配置および維持と関連したいくつかる利点が可能になる。

#### 【0039】

DMA120は、例えば販売後サービス、システム管理サービス、その他のサービスなどのアプリケーションおよびサービス提供物における積極的な関与を可能にする。DMA120を組み込んだ装置110は、システム管理アプリケーションおよびサービスで必要とされるいくつかの計算タスクを実行することができる。このアーキテクチャにおいて、例えばユーザ環境100に設置されたアプリケーション・サーバ200または供給業者300のバックオフィス・サーバ310と、ターゲット装置110とが協力して、システム管理提供物を完成させる。例えば図10、図13および図14に示したように、DMA120は、サービス実行環境124を提供し、そこで、サービス140は、サービス供給業者300のアプリケーション・サーバまたはホスト・システム310上で実行するシステム管理アプリケーションまたはホスト・システム320の全体または一部分として動作することができる。

#### 【0040】

DMAサービスは、装置のイベントを監視し、規定のアクションをとることができるこれが好ましい。DMA120は、関心のあるイベントが発生したときに加入者/ユーザにデータを発行できることが好ましく、また内部または外部クライアントあるいはユーザから指示されたときに装置110上に診断ルーチンなどのメソッドを呼び出すことができるこれが好ましい。これにより、装置固有の処理が、中央アプリケーション・サーバ320から装置110の近くに移される。アプリケーション・サーバ320の役割は、アプリケーション/サービス140の実行用の計算プラットフォームから、アプリケーション/サービス140の管理および構成に変換することである。したがって、装置110は、厳密なクライアント/サーバ・アーキテクチャにおける受動的データ・レポジトリと対称的にプロセスに積極的に関与する。

#### 【0041】

また、実施形態によるDMA120は、サービス140の動的な更新を実行し、エンド・ツー・エンド DCS プラットフォーム1内で動作する構成要素をサポートすることができる。DMA120を使用する装置110は、新しいサービス構成要素140を動的に追加することができる。これにより、ユーザあるいは装置110上に既にあるアプリケーション構成要素は、サービス140をサポートするためにそのような追加を要求することが

できる。また、これにより、必要に応じて、システムまたはDMAの再コンパイルまたは再起動なしに、構成要素の追加と削除を行うことができる。実施形態において、ターゲット装置110自体は、新しいサービスまたはアップグレードされたサービスの全体の追加あるいは既存のサービスの構成要素のサポートを開始する。これにより、本明細書に示したシステム1において、装置110は、装置110自体ならびにその上で動作するシステム管理サービスを維持する活動の開始を担当できるようになる。

#### 【0042】

実施形態は、さらに、開発者が一貫しかつ標準ベースのツール・セットで作業できるようにするアプリケーション／サービス実行環境124の必要を認識している。DMA120は、装置に依存しない販売後のアプリケーション140の開発を可能にする。DMA120インターフェースを使用して作成されたアプリケーション140は、新しいDMA対応装置またはアップグレードされたDMA対応装置に対応するように変更する必要がない。DMA120は、アプリケーション／サービスの装置独立性を達成するためにDMT Fによって採用されたモデル・ベースの手法を実施するが、この実施態様に、サービス・マネージャ126と呼ばれる新しい構成要素を追加する。サービス・マネージャ126は、主に、装置のために活動化された各サービス140のサービス140ライフサイクルを制御する役割を持つ。さらに、サービス140の制御と管理のために、サービス・マネージャ126は、サービス140を管理し、ローカルまたはリモートのシステム管理クライアントにプログラマチック・インターフェース(API)を提供することが好みしい。

#### 【0043】

エンド・ツー・エンドDCSプラットフォーム1内の動作において、DMA対応装置とDCSアプリケーション・サーバは、供給業者300のサービス・ホスト310上のアプリケーション・サーバまたはホスト・システム320によって、あるいはサードパーティ・サービス提供者によって、サービスの対応および管理を可能にする。DMA対応装置110とアプリケーション・サーバ320は協力して、動的対応を実現する。このシステムを使用すると、ユーザは、1組のアプリケーションを調査し、ユーザのニーズを満たすように1つまたは複数のアプリケーションを選択またはカスタマイズし、選択した1組のアプリケーションを注文することができる。続いて、アプリケーションを、インストールし、使用可能にし、起動し、監視し、かつ／または管理することができる。

#### 【0044】

新規および既存の装置の基準をカバーするために、実施形態におけるDMA120のアーキテクチャは、いくつかの形で配置を可能にする。例えば図4、図9および図11に示したような実施形態によれば、例えば、DMA120は、プリンタや多機能装置などのネットワーク接続装置110に組み込まれる。この実施形態において、DMA120は、ウェブ・サーバ側の構成要素になる。DMA120は、例えば、サービス構成要素をウェブ・サーバの後ろにホストするために、Javaサーブレットなどの標準的な方法を使用することができる。

#### 【0045】

代替として、DMA120は、複写機などの独立型の装置110またはDMA120を実行することができない現場の既存の装置の専用ハードウェア装置またはアドオン構成要素115に組み込むことができる。そのようなアドオン構成要素115を、図12、図16および図17に概略的に示し、後でより詳しく考察する。

#### 【0046】

もう1つの代替構成は、ネットワーク・アプリケーション220に、DMA120を、单一の装置プロキシ構成または複数の装置プロキシ構成として組み込むものである。そのようなプロキシ構成を、例えば、図7、図8および図11に例示する。

#### 【0047】

任意の装置製造業者にとって、販売後アプリケーションは、ユーザの忠実性を継続するために重要なことがある。記録装置および／または多機能装置の供給業者の場合、ドキュメント・システム装置の販売は、ユーザとの関係の始まりに過ぎない。継続的なサービス

10

20

30

40

50

、サポート、供給品補給および持続的保守が、ユーザの定着率を決定する主な検討要素になる。しかしながら、前に述べたように、販売後アプリケーションのコスト効率の高い開発、配置および管理自体に、いくつかの課題がある。複雑さは、開発者、装置製造業者、サービス所有者および顧客／ユーザを含む多数の利害関係者の存在によって高くなる。プラットフォームとして、DMA120は、実施形態に従って、すべての主な利害関係者をとりまとめ、すべての利害関係者の要件に対応するように設計されることが好ましい。

#### 【0048】

DMA120は、装置固有のインターフェース111とシステム管理アプリケーションの間に抽象化層を構成することが好ましい。例えば、図10、図13および図14を参照されたい。抽象化は、システム管理アプリケーションに対して、装置データ、イベントおよび操作の共通の見方を提供する。DMA120は、モデル・ベースの手法を使用して装置の抽象化を行う。このために使用される装置モデルは、DMTF（分散管理タスクフォース）コンソーシアムにおける業界標準活動に基づく。基礎として、強化したDMTF共通情報モデル（CIM）が使用される。しかしながら、装置インターフェースによる共通モデルの実施は独特である。販売後アプリケーション・クライアントとDMAの間の対話は、DMTF CIM Operations Over HTTPの仕様に基づく。

#### 【0049】

DMA120は、サービス・マネージャ構成要素126を追加することによってDMTF／CIMの実施を強化する。サービス・マネージャ126は、サービス140をロードし、DMA120に現在インストールされているサービス140のリストを維持し、サービス140の管理およびライフサイクルの制御を維持する役割を持つ。サービス・マネージャ126は、自動化されたプロセスとして動作することが好ましく、サービス140を、プログラムあるいはDMTFCIMAPI123でアクセスすることができる独立した構成要素として自動的に参照し開始することができる。サービス・マネージャ126は、装置110上のアクティブ・サービス140へのアクセスならびに1つまたは複数のアクティブ・サービス140の管理データを提供することができる。

#### 【0050】

DMAサービス・マネージャ126は、サービス・マネージャ126がロードするとき自動的に開始されるコア・サービスをサポートすることが好ましい。そのようなコア・サービスは、対応サポートを必要としないことが好ましい。また、サービス・マネージャ126は、DMA120の動的対応機能による実施を必要とする規定のサービス140をサポートすることが好ましい。

#### 【0051】

図15は、サービス・マネージャ126の起動と通常の実行を例示する概略的なフローチャートである。DMA120は、起動すると、サービス・マネージャを開始する（ブロック510）。次に、サービス・マネージャ126は、コア・サービスをロードし（ブロック511）、動的サービス対応ホストに問い合わせる（ブロック512）。サービス・マネージャ126は、さらに、サービス構成パラメータを解釈し処理し（ブロック513）、規定のサービス140をロードし開始する（ブロック514）。次に、サービス・マネージャ126は、管理のためにサービス140を監視し（ブロック515）、必要に応じてシステム管理クライアントと対話する（ブロック516）。

#### 【0052】

DMA120は、例えばSunのJava2マイクロ・エディション（J2ME）などの実質的にプラットフォームに依存しない言語を使用して作成されることが好ましい。したがって、DMAは、きわめて移植性が高く、Javaが使用可能な任意のシステム構成においてシステム構成要素として使用することができる。

#### 【0053】

実施形態による管理およびサービス・アプリケーション200は、ユーザの環境に配置することができる。いくつかのアプリケーション220は、例えば図7に概略的に示しているように、ネットワーク接続されているが装置サービスに自ら積極的に関与するよう

完全には使用可能にされていない装置 110 のサービス・プロキシとして部分的に挙動することによって、装置サービスの提供を可能にすることができる。換言すると、アプリケーション 220 は、DMA 120 あるいはサービス提供物 140 を直接サポートするソフトウェアを備えていないネットワーク接続された装置のプロキシとして働くことができる。例えば、Xerox (登録商標) 社の CentreWare Web (CWW) などのいくつかのアプリケーションは、シンプル・ネットワーク管理プロトコル (Simple Network Management Protocol) (SNMP) 対応の装置の装置プロキシとして働くことができる。装置内の SNMP エージェントが、提示されているサービスをサポートするために必要なデータと機能を提供するという点で、これは、装置をサービスの層にするための優れた方法のことである。そのような場合、CWW などのアプリケーションは、装置 110 の近くで動作するサービス 140 をホストする役割を負う。

10

#### 【0054】

サービス・ホスト 310 などのエンド・ツー・エンド・システム 1 のバックオフィスまたはホスト部分 300 にとって、装置 110 は、アプリケーション 220 によって代理されたときに、装置がサービス・ホスト 310 と直接通信した場合とほとんど同じように見える。代理の装置と直接の装置との違いは、様々なレベルのファームウェアを有する直接装置と関連した違いほど実質的な差がない。装置の機能の違いは、装置ベース・サービスの対応システムによってバックエンド／サービス・ホスト 310 上で管理することができる。そのような違いの影響は、所定の装置のきわめて特殊な機能を利用する高度なサービスが、ほとんど移植性をなくすことであり、すなわち、CIM 拡張機能の製品固有の部分のために作成されたサービスは、まったく移植性がなくなり、他の装置と互換でなくなることがある。しかしながら、コアおよび共通モデルに作成されたサービスは、移植性を維持し、配置の問題は、対応システムによって管理される。

20

#### 【0055】

アプリケーション 220 は、また、直接ネットワーク接続されていないがアドオン接続オプションを有する装置のサービス・プロキシとして挙動することができる。そのような接続オプションは、イーサネット (登録商標) などのハードワイヤードでもよく、あるいは Bluetooth (登録商標) や IEEE 802.11 などの無線でもよく、その有効範囲が局所的でも広範囲でもよい。例えば、印刷用にネットワーク接続されてない独立した複写機は、802.11b や他の無線ネットワークなどの小規模無線 LAN 接続が追加される。プロキシは、DMA のないネットワーク接続装置が代理されているときとまったく同じに挙動するが、また、プロキシは、アドオン接続機能が付加された装置と通信するために使用される無線アクセス・ポイントに必要なハードウェアを含む。そのようなシステムの例は、LAN 上で物理的にネットワーク接続されまた無線アクセス・ポイントが付加されたサーバ上にインストールされた CWW である。バックオフィス・エッジ・ホスト 310 にとって、無線式に代理された装置 110 は、LAN 上で代理された装置と違いがないように見える。

30

#### 【0056】

アプリケーション 220 は、また、ユーザ環境内のサーバからサービスをひとまとめに管理することを可能にする。実施形態において、装置プロキシは、代理装置と関連したサービスの集約およびグループ管理機能を提供することができる。これは、例えば、システム管理者またはユーザが 1 組の装置上のサービスのステータスを調べるグラフィカル・ユーザ・インターフェース (GUI) である。

40

#### 【0057】

サービス・ホスト 310 などのバックオフィス・システムに対するサービスおよびそのトランザクションの共通装置インターフェースは、装置自体で使用されるインターフェースの拡張部分でよい。これにより、サービスは、装置が直接の状況と、装置が代理を使用する状況の両方で動作することができる。プロキシと関連したトランザクションおよびデータを処理する限られた組の特定の装置プロキシ拡張機能と共に、サービス・ホストと直接通信する装置からの API がサポートされる。装置に向けられたトランザクションはすべて

50

、同じように見えなければならない。

#### 【0058】

図2を参照して、例示的な実施形態の追加部分を示す。サービスのホスト部分、すなわちサービス・ホスト310は、ユーザ・サイト100、200の外にあってもよく、製造業者や他のサービス提供者の設備300にあってもよい。装置サービス・エッジ・ホスト410は、現場に配置された装置サービスのトランザクションとサービス管理を処理する。これは、分散サービス140をサポートするメッセージ待ち行列の管理とソフトウェア・モジュールおよび構成パラメータの対応を含む。エッジ・ホスト410は、また、セキュリティのホスト・エンドと装置サービス・システム1によって使用されるサービス・モデルの役割をする。

10

#### 【0059】

エッジ・ホスト410は、また、サービス・スponサ・システム310への接続を提供する。これは、外部装置サービス界(external device services world)を内部(またはサードパーティ)サービス界(internal (or third party) services world)に接続する。サービスのスponサとなる各組織の接続は、エッジ・サーバによって維持され、サーバとホストを維持するエンティティのセキュリティ・ルールおよび規則に従うことができる。エッジ・ホストによって、装置110または装置プロキシ220が、サービス・ホスト310上に含まれるサービスのバックオフィス集合体との対話の詳細の直接的な知識を持つ必要がなくなる。これは、装置サービスを、管理しやすくかつ頑強に配置する際に有利である。実施形態は、基本的な操作を標準化するために、バックオフィス・スponサ組織に共通サービスAPIを提供することが好ましい。特定のサービスは、所定のアプリケーションのトランザクションの内容を拡張しカスタマイズすることができる。

20

#### 【0060】

引き続き図2を参照して、ユーザの環境100、200に、装置110と、CWWなどのユーザ・アプリケーション220を配置することができる。これは、管理されたサービス環境でもよく、通常のユーザ環境でもよい。インターネットや他の適切な長距離接続機構400を横切って、ユーザの内部システムおよびネットワーク100、200から、サービスおよび通信が分散されお互いをつなぐ。周知のウェブ・サービスならびに将来のウェブ・サービスは、装置/アプリケーション110および220とエッジ・ホスト410との間、ならびにエッジ・ホスト410と内部サービス提供者310の間で使用されることが好ましい通信機構130、230、330を含む。システム1は、ユーザ側(ファイアウォール160)とサービス提供者側(ファイアウォール360)の両方の既存のファイアウォールによって動作する能力のために、基本的なIT産業やその他の規格を満たすように構成されることが好ましい。このシステムを適切に動作させるために、ユーザのファイアウォール160の特別な構成は必要ない。

30

#### 【0061】

サービス供給システム300は、すべてのユーザ・システムが使用する頑強で十分に管理された24x7レベルのサービスおよび障害回復を提供する基本的な供給業者インフラストラクチャの一部であることが好ましい。前に示したように、エッジ・ホスト410は、現場に配置された装置サービスのトランザクションおよびサービス管理を処理することができる。これは、分散サービスをサポートするメッセージ待ち行列の管理ならびにソフトウェア・モジュールおよび構成パラメータの対応を含む。これは、また、装置サービス・システムによって使用されるセキュリティおよびサービス・モデルのホスト・エンドの役割をする。実施形態において、エッジ・ホストは、また、外部装置サービス界を内部(または、サードパーティ)サービス界に接続する。サービスのスponサになる各組織への接続は、エッジ・サーバによって維持され、供給業者のセキュリティ・ルールおよび規則に従うことが好ましい。

40

#### 【0062】

実施形態において、エッジ・ホスト410によって、装置または装置プロキシは、含まれるサービスのバックオフィス集合体との対話の詳細の直接的知識を持たなくともよくな

50

る。これは、装置サービスを管理しやすくかつ頑強に配置するのを助けることができる。基本的な操作を標準化するために、共通サービスA P Iが、バックオフィス・スポンサ組織に提供される。特定のサービスが、所定のアプリケーションのトランザクションの内容を拡張しカスタマイズすることができる。

#### 【0063】

##### 「多重送出パス」

最高レベルにおいて、システム1は、システムによって、サービス140が装置110とバックオフィス（供給業者）300間で直接動作したり、装置プロキシ220の支援により動作したりするように設計される。これにより、ユーザが直接使用可能にされる新しい装置をゆっくりと取得している間に装置プロキシ220がきわめて迅速に多数のレガシ装置をサービス提供物にすることができるので、最も広い配置ができる限り迅速に行われる。また、ユーザによっては、プロキシを、個別に各装置からではなくそのサイトから出るメッセージのコンソリデータ／クリーニング・ハウスとして挙動させるのを好む場合があるので、両方の動作モードを有することも重要である。他のサイトでは、ユーザは、装置プロキシをインストールすることを望まない場合があり、したがって、サービスを直接使用可能にする必要がある。それぞれのパスを使用可能にする他に、最後に、ユーザが両方のシナリオを同時に行うことができるため、サービスを一緒に機能させることは有益である。

#### 【0064】

複数のパスが、配置のフレキシビリティを高めることができるように、そのようなパスを、サービス提供者の視点から見えなくすることは有益である。実施形態は、装置110とプロキシ220をできるだけバックオフィス・システム310から分離することが好みしい。これらの2つの強力な抽象化と分離により、装置110またはバックオフィス300内の機能を階段的に独立に配置させることができる。さらに、どちらかの端にあるシステムの変更を行う必要がある場合、変更は、適切な抽象化が使用可能にされた場合にシステム1全体に影響を及ぼさず、保全性が強化される。

#### 【0065】

図10を再び参照すると、実施形態の全体にわたる抽象化は、装置レベルで、D M A 120に組み込まれた抽象装置モデル122を含む。抽象装置モデル122は、基礎としてD M T F C I Mを使用して構成されることが好みしい。実施形態において、装置モデル122とプラットフォーム124は両方とも、D M A 120にある。共通サービスは、提示されるサービスに関係なく供給業者のドメイン300を指定する。また、バックオフィス／供給業者レベル300で、実施形態は、サービス・スポンサの共通A P Iを使用して、供給業者バックオフィス300からのサービスを構築し管理する。共通A P Iは、タイプまたは接続機構に関係なく、装置110を同じように処理する。

#### 【0066】

実施形態におけるこのアーキテクチャは、供給業者への直接の装置通信またはプロキシを介した通信の観点から、配置のフレキシビリティなどのフレキシブルな配置オプションを提供する。フレキシビリティの他に、サービス自体を、その多数のパラメータをカスタマイズできるように定義することができる。このサービス・カスタマイズには、例えば、リモート監視サービスの一部分として送られるデータや、データ読み取りを供給業者に送る時刻または頻度がある。正確な構成パラメータは、提供されるサービスに固有でよい。

#### 【0067】

実施形態のプラットフォームは、サービスの構成を容易に管理できるように設計されている。このシステムにより、ユーザ・サイトにあるすべての装置に関して、あるいはある場所に関係なくユーザが所有するすべての装置に関して、サービスの構成を、個別の装置通し番号で指定することができる。実施形態において、この管理は、サービス提供者によって制御されるバックオフィスで行われる。

#### 【0068】

フレキシブルな配置オプションのさらに他の部分は、図3に概略的に示したように、実

10

20

30

40

50

施形態に従って、装置に利用可能なサービスに加入基準を使用することである。加入プロセスは、例えば、個別のサービス提供者によって制御され管理することができ、任意の所定の装置に提示されるサービスは、ユーザの要望とサービス提供者の認証を組み合わせることによって制御することができる。したがって、同じ製品系列であっても、すべての装置が、いつでも同じサービスを提供したりインストールしたりしなければならないとは限らない。

#### 【0069】

DCSと関連するいくつかのアクティビティがある。例えば、Axeda、Embrace Networks、Questra、およびImaging Portalsは、サービス面で有効であった。これらの技術的な実施態様の例は、Embrace Networksの特許出願の付与前公報番号2002-0133581A1であり、この出願は、参照により本明細書に組み込まれる。しかしながら、従来技術には、対応態様がなく、また対応のための重要なサポートがあるとは思われない。

10

#### 【0070】

4th passなどのいくつかの会社が、汎用対応ソフトウェアを販売しているが、従来技術はどれも、本発明の態様を包含するとは思われない。参考のために、Sunは、<http://java.sun.com/j2ee/provisioning/industry.html>にそのような対応ソフトウェアの一般的なリストがある。さらに、すべての会社が、携帯電話産業をそのターゲット市場として狙っていると思われる。

20

#### 【0071】

前述のように、広域的な電気通信会社が、携帯電話上のサービスを提供し始めている。これを達成するために、すべての携帯電話は、CLDCと呼ばれるJava規格を使用している。このリリースされた規格は、携帯電話などの小型装置上でJavaプログラムがどのように動作できるかを規定し、またはより重要なことには、Midletsと呼ばれるモジュラ・プログラムを、実行時にCLDC Java環境にどのように追加できるかを規定している。

20

#### 【0072】

この規格は、対応ユニットと、それが装置側でどのように受け入れられ組み込まれるかを定義しているが、サーバの側面については何も言及していない。このために、電気通信会社は、自分自身の対応サーバ・ソリューションを作成するか、前に列挙した対応ベンダーからのものを購入していた。この分野での高い競争環境のために、代替ソリューションについて検査する方法はない。

30

#### 【0073】

第2の関連する規格は、OSGiと呼ばれる。OSGiは、ローカル・ネットワーク接続された装置が、リモート・サーバと通信して、モジュラ・サービスをダウンロードし実行できるようにするJavaを利用したリリース規格である。この規格は、CLDC/Midletsと比べて、業界であまり支持を受けてこなかった。OSGiは、また、対応のサーバ側の問題を避けている。

#### 【0074】

第3の規格は、SyncML Device Managementである。SyncMLは、モバイル装置をいくつかのサーバ・ベースのソースと同期させる詳細に焦点を当てたリリース規格である。この規格の狙いは、カレンダや約束のようなものにある。昨年、この同期プロトコルは、モバイル装置上のサービス設定を明示的に変更し、そのモバイル装置にサービスをダウンロードできるようにする装置管理努力によって拡張された。SyncMLは、サーバ側の対応の問題を避けている。

40

#### 【0075】

最後の規格は、名前はないが、一般にJSR-124と呼ばれる。要するに、Javaプログラマは、Java Community Process (JCP)を使用して、Java Specification Request (JSR)を、Java言語への追加および拡張機能として作成し規格化する。JSR-124は、J2EEクライアント対応仕様 (J2EE Client Provisioning Specificati

50

on) である。J 2 E E は、ハイエンドなトランザクション処理で J a v a を使用するための規格である。この周辺の市場は大きく成長した。実際には、J S R - 1 2 4 は、対応システムを表現するフレームワークを定義しようとしている。ほとんどすべての対応する新規企業と電気通信会社はほとんど、J S P の会員である。これは、すべての対応システムが J 2 E E システムとすべて標準的な方法で対話できるほど一般的でかつ、ベンダが代替の競合ソリューションを作成できるほど緩いものになるようにする。規格は、公開され草案が評価されている段階である。

#### 【0076】

実施形態には、共有ユーザ・サービス・ライフサイクルに基づいた共通対応モデルの定義と実施が含まれる。Provisioning Server (PS) 3 1 0、P S に話しかけるD C S 装置 1 1 0、およびD S C 装置と対話する供給業者は、対応プロセスが動作する方法に関する共有モデルに従って動作することが好ましい。P S 3 1 0 と対話する関係者ごとに、役割と責任を定義するライフサイクル・モデルを作成することができる。役割と責任に基づいて、関係者がその役割に基づく目標を達成できるように、文法とコマンドが作成された。

#### 【0077】

例えばこの節のすべての要件を満たすサービス・ホスト 3 1 0において動作する対応サーバ 9 0 0 のアーキテクチャと実施様を、例えば表 1 と図 2 0 に概略的に示す。図 2 0 において左から右に進み、最初の主要モジュールは、サービス顧客インターフェース 9 0 1 である。これは、外部ユーザ 1 1 0 と外部装置 2 2 0 とのすべての対話の担当することができる。また、装置と顧客が使用する可能性がある様々なプロトコルから他の P S モジュールを分離することが好ましい。実施形態での好ましいプロトコルは、ウェブ・サービスであるが、将来的には、h t t p 、電子メール、携帯電話やその他の送信フォーマットに拡張することができる。入ってくるトランザクションの場合、要求を処理するのに適正な内部リソースにトランザクションを送る。出てゆくトランザクションの場合、装置またはユーザのために待ち行列に入れられた他の P S モジュールの出力を取得し、それを、装置または顧客と対話するのに必要なプロトコルに変換する。

#### 【0078】

10

20

【表1】

アクター	定義	主な役割と責任
サービス・スポンサ	PS900上のサービスの作成および配置をサポートする供給業者や他のパーティ内の組織	<ul style="list-style-type: none"> <li>サービス開発者を使用して、サービスのPSおよびDCS装置部分の必要コードを開発し配置する。</li> <li>サービス・スポンサから出されるコマンドがIMシステムのローカル状態と同期するようローカルIMシステムとPS900の間の提携を行う。</li> <li>PS900に指示して、特定装置のサービスをローカルIMと同期して使用可能にし使用不能にする。</li> </ul>
サービス開発者	DCSベースのサービスを実施するコードを開発する。	<ul style="list-style-type: none"> <li>DCSガイドラインを使用してコードを開発する。</li> <li>サービスを構成するコード・バンドルをPS900に配置する。</li> <li>関連プラットフォームやサービスに関する他のパラメータ情報を識別することによりPS上のサービスを定義する。</li> </ul>
ポリシおよび基本設定セッタ	サービス配置のすべての様子を制御する「規則」を定義する。	<ul style="list-style-type: none"> <li>次のものを定義する「規則」を開発する。           <ul style="list-style-type: none"> <li>サービスのデフォルト・パラメータの構成方法。</li> <li>サービス・パラメータ情報および／またはIMシステムから得られたサイト、顧客、地理的単位または他のグループのパラメータの均一さ。</li> </ul> </li> </ul>
外部装置	ユーザ・サイトにあり、直接あるいはDCSインターフェースを提供するローカルまたはリモート・プロキシ・サーバを介してアクセス可能なDCS互換装置。	<ul style="list-style-type: none"> <li>PSを使用してサービス・ライフサイクルの変更（追加、削除、修正、更新など）を通知する。</li> <li>PSを使用してローカル破局的故障に基づいて復元要求を送る。</li> <li>関連する装置の構成変更をPSに知らせる。</li> <li>関連するイベントまたは状態の変更をPSに知らせる。</li> </ul>
内部ユーザ	サービスの実行に関与する供給業者組織。	<ul style="list-style-type: none"> <li>サービスに関連したトランザクションを、PSと場合によっては他の装置または他のユーザに送る。</li> <li>同じように装置からトランザクションを受け取る。</li> </ul>
外部ユーザ	サービスの実行に関係する供給業者ではないユーザ	<ul style="list-style-type: none"> <li>サービスに関連したトランザクションを、PSと場合によっては他の装置または他のユーザに送る。</li> <li>同じように装置からトランザクションを受け取る。</li> </ul>

表1

10

20

30

40

## 【0079】

エンティティ管理モジュール902は、エンティティ情報をローカライズしサーバ900の他の部分から分離することが好ましい汎用PSリソースである。このモジュールは、装置、ユーザ、その基本設定、関連する位置情報などのエンティティに関する情報を保持する。ローカルでないエンティティ情報では、エンティティ管理モジュール902は、そのような他のIMシステムと接する1つのポイントである。モジュール902は、ローカルおよびネットワーク・ベースの情報にシームレスなインターフェースを提供する。

## 【0080】

50

注文処理モジュール (order processing module) (O P M) 9 0 3 は、サービス・スポンサからのまたポリシおよび基本設定モジュール (policy & preferences module) (P P M) 9 0 4 によって作成された注文の処理を指示する役割を持つ。O P M 9 0 3 は、注文要件を達成するために必要なP S モジュールと対話する。O P M 9 0 3 は、また、スポンサからの照会に応えることができるよう注文の状況を追跡することができる。

#### 【0 0 8 1】

登録、認証および許可モジュール (registration, authentication, & authorization module) (R A A M) 9 0 5 は、システムのセキュリティを常に維持する役割を持つ。R A A M 9 0 5 は、P S のすべてのユーザに権限を与え、ユーザが特定のトランザクションを実行できるようにすることができる。内部ユーザと外部ユーザをすべて適正に登録する役割を持つ。R A A M 9 0 5 は、エンティティ・モジュール 9 0 2 と一緒に作業して必要な情報を得ることによってこれを行う。R A A M 9 0 5 は、また、サービス顧客モジュール 9 0 1 と注文処理モジュール 9 0 3 と一緒に作業してトランザクションのセキュリティに関連するアーティファクトを分離する役割を持つことができる。

#### 【0 0 8 2】

サービス定義モジュール 9 0 6 は、P S 9 0 0 から提供されるすべてのサービス 1 4 0 に関するすべての定義情報を維持する役割を持つ。含まれる情報の例は、バージョン情報、ファイル構成、サービス相互関係、製品系列サポートである。

#### 【0 0 8 3】

サービス開発者インターフェース・モジュール 9 0 7 は、サービスを開発し、分散し、更新するサービス開発者の作業を支援する役割を持つ。サービス参加者インターフェース・モジュール 9 0 8 は、すべてのユーザと対話して、サービス・ライフサイクルおよびサービス・トランザクション情報を適正なリソースに導く役割を持つ。

#### 【0 0 8 4】

実施形態は、一般的なソリューションとして例えば規則や制約などのソフトウェア計算技術を利用して、サービス・ポリシをフレキシブルにモデル化し、開発し、検査する。全体的に、対応決定自体は、それほど重要ではない。すなわち、サービス 1 4 0 を必要とする装置 1 1 0 の場合、P S 9 0 0 は、許可されるかどうか、装置 1 1 0 の動作パラメータ情報（モデル・タイプ、O S バージョンなど）と互換性があるバンドル（インストールされるサービスを構成するコード・ファイルの集まり）があるかどうか、複数のバンドルがある場合にそのうちのどれを選択すべきか、またサービス 1 4 0 のパラメータ設定値（ある場合）が何でなければならないかを決定する。一般に、実施形態において、以上の問題を解決するために使用できる「ビジネス・ルール」を実現するコードを作成することはできない。コーディングは、規則の変更ごとに必要になり、規則は、ポリシ作成者によって直接調査されず、それぞれの問題が他のものと分離可能であると仮定する。さらに、上記のすべての問題の答えを決定する1つのポリシ作成者がいると仮定する。したがって、実施形態において、代替のソリューションが提供されなければならず、また提供される。

#### 【0 0 8 5】

適切な制約または規則システムの導入により、有利な恩恵が提供される。コーディングは、「規則」がより高い抽象化レベルで入力されたときに大幅に減少する。さらに、規則は、コンピュータまたはプログラミングに満足できないポリシ作成者によって調査可能である。さらに、それぞれの問題に関係する制約および規則として実現される知識を容易に組み合わせることができ、分離可能性について心配しなくてもよい。制約と規則は、以上の問題の結論に関する多数のポリシ作成者の存在をサポートする。価値連鎖に関する様々なグループに基づく規則と制約の干渉は、より容易に識別し解決することができる。

#### 【0 0 8 6】

状況によっては、ポリシ・ベースの知識を使用して正確なバンドルとパラメータを識別する対応サーバ 9 0 0 の能力が有利なことがある。例えば、この能力は、P S 9 0 0 が追加サービス要求を受け取り、前述の問題に対する答えを計算する必要があるときに適用されることが好ましい。さらに、P S 9 0 0 は、この能力を、任意のサービスのポリシ作成

10

20

30

40

50

者がポリシ知識を更新したときに使用する。PS900は、PS900に積極的に接続された既存の関連装置への変更、追加、または削除の影響を計算することができる。次に、PS900は、変更の目的を達成するために影響を受けた装置110に対して必要な変更要求を生成して、将来のすべての追加サービス・トランザクションの変更を使用することができる。PS900が、装置110からの構成変更の通知を受けると、PS900は、その変更のために装置のサービス140および／またはパラメータを変更しなければならないかどうかを決定する。必要は場合は、PS900は、ポリシ知識からの要求に応じて装置110の変更要求を生成することができる。

#### 【0087】

ポリシ設定者は、規則の使用により、内部顧客要件または外部顧客要件に基づいて均一なサービス・バージョンまたはパラメータ設定値を定義することができる。この均一性は、ユーザ・レベル、サイト・レベル、装置カテゴリ、または他の関連グループで定義することができる。

10

#### 【0088】

要するに、サービス加入および配置方法は、対象となるサービス提供物140のユーザまたはユーザDMA120による識別と、そのようなサービスの活動化の要求を含む（ロック501）。エッジ・ホストによる計画的な確認の間、あるいはその目的のための特別な接続の間、DMA120は、供給業者システム300に、対象および要求された活動化に関するメッセージを送る。供給業者システム300は、エッジ・ホスト410からメッセージを取り出し、ビジネス・ルールおよび作業プロセスを適用してユーザ適格性を決定する（ロック502）。ユーザが、承認された場合、供給業者システム300は、エッジ・ホスト410に、要求されたサービス140を追加できることを通知する（ロック503）。次に、DMA120は、エッジ・ホスト410に連絡して、サービス140を追加することができるというメッセージを受け取る（ロック504）。次に、DMA120は、サービス140を作動させ、必要に応じて、そのサービスをダウンロードしつ／またはインストールする（ロック505）。次に、新しいサービスが、配置され、動作する（ロック506）。

20

#### 【0089】

サービスの販売は、複数のチャネルで行うことができる。このプロセスは、スポンサ組織（サービス供給業者）によって所有されることが好ましく、スポンサ組織が選択した方法で行われる。これは、例えば、必要に応じて装置から行うことができる。

30

#### 【0090】

特定のユーザが所定の装置上のサービスを使用可能にしたいことがスポンサ組織に通知された後、実施形態は、スポンサ組織が、特定のサービスにとって妥当なビジネス・モデルに従わなければならないビジネス・ルールおよび請求書発行／納品書発行プロセスを適用することを規定している。スポンサ組織が、指定されたサービスをユーザに提供することを装置に許可できると決定した場合、スポンサ組織は、エッジ・サーバ上の共通サービス注文／エントリAPIを使用して正式に注文を出す。これは、実施形態において、所望のサービスの配置および構成を開始することができるメッセージを生成することができる。

40

#### 【0091】

メッセージは、送出のために待ち行列に入れられることが好ましく、プロセスは、メッセージが送出されるまで待つ。要求装置または装置プロキシが、注文メッセージを取得した後で、システムが構成され、必要に応じて追加のソフトウェアがダウンロードされ、新しいサービスが開始される。サービス・スポンサは、実施形態によるシステムにより、サービス・スポンサが決定するとの基準が必要かに基づいて、必要に応じてサービスをオン・オフできることが好ましい。サービスは、装置に依存しないように作成されることが好ましい。装置モデル・エージェントによって提供される共通情報モデルは、実施形態において、共通データおよび方法の装置に依存しない装置を提供する。すべてのユーザが同じ要件を持つとは限らないので、サービスは構成可能である。構成可能なサービスは、必要

50

とされる可能性のある要件および操作の変化に対応することを可能にする。サービスは、現地に既に配置済みの装置によって新しいサービスをユーザに迅速に配置できるように動的にロードすることができる。また、サービスは、最初に配置された後で管理を可能にするライフサイクルを有する。ライフサイクルのトランザクションの例には、サービスの追加、サービスの削除、サービスの修正、サービスの同期、装置登録、およびプロキシ登録があるが、これらに限定されない。

#### 【0092】

実施形態において、DMA120は、共通に配置されたサービス140において、装置110の組み込まれた計算能力、データ、および機能にアクセスし使用できるように定義される。組み込まれたエージェント122とサービス・プラットフォーム124により、実施形態は、システム1全体に入るサービス140のローカル動作をサポートすることができる。これは、サービス提供物のサポートにおいて、共通接続性、サービス・マネージャ、共通データ・アクセスおよび方法、およびサービス提供者／供給業者への確実な通信を提供する。

#### 【0093】

前述のシステム、構成要素、方法および実施形態の場合、システムを配置することができる方法はいくつかある。この配置のフレキシビリティは、このシステムの大きな利点であり、構成要素の詳細な設計およびシステムが従う挙動モデルと密接な関係がある。抽象化とモジュール性が定義された実施形態によって、そのような配置オプションをすべて同時にインスタンス化することができる。多くのユーザ装置において、完全な有効範囲を保証するために複数のオプションを配置することが可能となる。図4から図9と図11は、本発明によるシステムの可能な配置オプションを表すいくつかの例示的な実施形態を示す。

#### 【0094】

図4に示した例示的な実施形態の配置Aは、Xerox（登録商標）社などのいくつかの会社から現在出荷されているスマート装置の好ましい実施形態である。これは、サービス140の配置をサポートするためにユーザに要求されるインフラストラクチャの量を制限することができ、最も単純な実施態様を提供する。ユーザの環境では追加のハードウェアまたはソフトウェアを導入する必要はないが、装置110は、サービス・プラットフォーム124を含むDMA120の機能を備えていなければならない。この実施形態は、装置ソフトウェアをアップグレードするか、別の方法を使用して現場の装置にDMAとサービス・プラットフォームを提供しない限り、現地に既にある多数の装置に対応することはほとんどできない。DMAを介した装置とバックオフィス・ホストとの間の通信は、実質的に物理的媒体から独立しているが、好ましい実施形態は、ユーザのネットワークおよびユーザのインターネット・アクセスを使用して、供給業者のホスト・システムに接続する。当然ながら、例えばローカル無線、長距離無線、電話、無線電話、衛星電話などの他の通信方式を使用することもできる。

#### 【0095】

図4に示したように、各装置110は、それぞれのDMA120を含み、DMA120によって促進される自分のサービス層141で自分のサービス140を実行する。管理およびその他のアプリケーション220は、ユーザの環境100にあってもよくあるいは他の場所にあってもよい別の装置200上で使用することができる。装置110は、HTTP、HTTPPS、SOAPなどのウェブ・サービス250を使用して、供給業者300およびその中のサービス・ホスト310と通信するが好ましい。サービス・ホスト310は、サービス320と、適切なときにDMA120からの通信を評価しサービス140を配置することができるホスト・システム340を含む。

#### 【0096】

図5に示した別の例示的な実施形態の配置Bは、既に現場にある装置と、必要な技術が組み込まれていないサードパーティ製の装置が、装置サービスをサポートできるようにする。そのように複数の装置を処理することができるが、この説明は、簡略化するためにそ

10

20

30

40

50

のような1つの装置に注目する。この場合、比較的小さなアドオン構成要素115が、装置110に追加される。アドオン構成要素115は、アドオン構成要素115が内部データと装置110の機能にアクセスできるようにするために必要なソフトウェアおよびDMA120ならびに装置110への1つまたは複数の接続を含む。アドオン装置115が接続された場合、装置／アドオン構成要素の組み合わせは、サービス・インフラストラクチャおよびバックオフィス・システムの他の部分からは、図4に示した配置Aのように、完全に使用可能にされた装置のように見える。これは、レガシおよびサードパーティ製造設備の実施形態による装置サービスを提供する。この場合、アドオン構成要素115は、DMA120とそれに付随するサービス環境124により、図4の配置Aのようにウェブ・サービス250を介して供給業者300と通信する。

10

## 【0097】

図6に示した第3の例示的な実施形態の配置Cは、アプリケーション220が、装置110のうちの少なくともいくつかのサービスを実行するプロキシとして動作することができるプロキシ構成を使用する。必要なソフトウェア成功因子のない装置110に、DMA120やプラットフォーム124などが組み込まれている。しかしながら、装置のサービス・プロキシとして働くアプリケーション220は、例えばLAN、電話、無線、その他の通信媒体などの装置110と通信することができる。基礎的なプロキシは、選択された1組のサービス140のサービスAPI140を実現するが、完全なDMA120と装置110自体への標準的な動的サービス配置方法を使用しないことが好ましく、それは、そのような機能をレガシ装置でサポートすることができないためである。この配置は、また、単純なプロキシと装置との間の接続の豊かさによって制限され、データまたは機能にリモートでアクセスできない場合には、それらを必要とするサービスを配置することができない。

20

## 【0098】

図7に示した第4の例示的な実施形態の配置Dは、プロキシ構成のより有利な形態である。この実施形態は、必要な組み込まれたソフトウェア成功因子（すなわち、DMA120）なしに装置を使用可能にするが、例えばLAN、電話、無線などの他の方法で通信して、サービス配置システムに関与することができる。装置110は、装置110のサービス・プロキシとして働く1つまたは複数のアプリケーション220と通信する。サービス・プロキシは、サービス・プロキシと通信する各装置110のDMA120をホストすることができるDMA対応プロキシである。さらに、サービス・プロキシは、通信する装置110のDMA120を管理することができる。これにより、サービス140は、サービス・プロキシ上で、サービス140が装置110自体の上で直接動作しているかのようにまったく同じように動作することができる。また、これにより、追加のローカル・アプリケーションを、DMA120ならびに各システムのデータおよび機能の共通情報モデル表現を利用することができるサービス・プロキシ上に作成することができる。これは、アプリケーションを大幅に単純化することができ、それは、アプリケーションを、各装置に固有の実施態様から隠すことができ、CIM内のデータと方法の共通表現に構成するだけよいためである。これは、CIMとDMAに対して作成されたときにサービスが獲得するのと同じ利点である。さらに、CIMとサービスの複数のインスタンスを管理することができるDMAの一部分を、一旦インスタンス化し、複数の装置のDMAプロキシを管理するために使用することができる。すなわち、すべてのプロキシ化装置に関して完全なDMAを複製する必要はなく、これにより、この実施形態が、接続された装置のすべてのDMAを1つのサーバ上に単純にドロップするよりも効率が高くなる。

30

## 【0099】

サービス・プロキシの実施形態のもう1つの態様は、CIMとサービスの複数のインスタンスを管理することができるDMAの部分を一元化インスタンス化し使用して、複数の装置のDMAプロキシを管理することができる。したがって、すべてのプロキシ化された装置について完全なDMAを複製する必要はなく、むしろ、複数の装置に1つのDMAを使用することができる。これにより、1つのサーバ上に各装置ごとの1つのDM

40

50

Aを単純にドロップするよりも、配置の効率が高まる。

#### 【0100】

特定のバージョンの配置CおよびDにおいて、実施形態は、パーソナル・コンピュータに直接接続されたプリンタの装置プロキシの配置をカバーする。プロキシをユーザのコンピュータ上にホストすることができ、また、プロキシが対話する装置は、パラレル・インターフェースを介して接続されたプリンタなどのプリンタである。実施形態において、プロキシは、DMAまたはサービス・インターフェースを実装するデータの追加ソースとして、直接接続されたプリンタの印刷ドライバに接続することもできる。コンピュータは、DMAをホストすることができ、装置への直接接続と印刷ドライバや他のアクセス機構を介したローカル装置によってサポートされるという点で、直接接続されたプリンタは、サービスおよびシステム管理の観点からネットワーク接続されているように見える。

#### 【0101】

図8に示した第5の例示的な実施形態の配置Eは、図6と図7に示した例示的な実施形態の一部分を変形したものを含む。サービスは、インターネットで広がる実施形態と類似の形で、ローカルに、すなわち実質的に自己完結的なサイト内で提供することができる。そのような実施形態は、DMA120の抽象化を使用して、ローカル装置110へのより一貫した管理とサービスを提供する実施態様を可能にする。これには、バックオフィス・サービス提供者300への接続がないが、サービス140は、セキュリティのために、ユーザに固有でもよく単純に自己完結されてもよい。この場合、ローカル・サービス140および装置110の管理を、すべての装置110の集中的な場所からより現地化されたドメインに移すことができる。ユーザは、例えば、そのような実施形態において、必要に応じて、アプリケーション・サーバと、ユーザのセキュリティ要件によってエッジ・ホストとを含む、バックオフィスの等価物をそのインターネット上で動作させることによって、供給業者の役割を負うことができる。これは、サードパーティによって提供された場合にそのようなシステムの維持とサポートの複雑さを高めるが、抽象化が定義された場合には有用な構成が可能である。

#### 【0102】

さらに例示的な実施形態の図9の配置Fは、使用可能にされた装置110から通信を受け取る複数のアプリケーション・サーバ310および／または複数のエッジ・ホスト410を使用可能にする。配置Fは、例えば配置A、B、DおよびEの要素を結合した実施形態である。サービス140は、適切なエッジ・ホスト410を介して適切なアプリケーション・サーバ310と適切な方法で連絡するためにサービス140に必要なすべてのものを示すように作成することができる。さらに、エッジ・ホスト410が装置110を接続するサービス・ホスト310は、任意の特定のサービス・ホストまたは供給業者300に限定されず、エッジ・ホスト410から提供されるサービスAPIが接続を可能にする限り、サービスを提供する適切などの関係者にも接続することができる。

#### 【0103】

##### 「装置モデル・エージェント」

例えば図10の概略図において、前に考察し示したような装置モデル・エージェント(DMA)120は、実施形態によるエンド・ツー・エンド・システム1の特権を与える構成要素である。DMA120を、装置110、アドオン・モジュール115および／または装置／サービス・プロキシに組み込んで、サービス140が動作する共通装置モデル122、CIM API123および装置サービス環境124を提供することができる。DMAの役割は、その寿命全体にわたって装置を取り巻く業務プロセスおよびサービスに積極的に関与する能力を装置110に提供することである。これは、分散管理タスクフォース(DMTF)からの共通情報モデル・オブジェクト・マネージャ(Common Information Model Object Manager)(CIMOM)の態様と、組み込まれた動的なサービスの操作および管理の新しい環境の態様とを組み合わせる。エージェントは、サービスをローカルに操作し、CIMにおいて提供された情報を管理する役割を持つ。エージェントは、装置、サービス(ローカルなサービスと、ネットワーク接続された環境の全体にわたって分散

10

20

30

40

50

されたサービスの両方）、および他の分散されたシステム構成要素と対話する。

#### 【0104】

DMAは、DMT Fによって指定されているような装置と独立CIM APIを提供するが、装置と独立のサービスAPIも提供する。ソフトウェア・エージェントのように、DMAは、ローカルに行われるかまたは他の分散された構成要素との対話によって行われる自主的挙動と適応的挙動で連動することができる。DMAは、また、例えば、この場合も装置および環境におけるイベントにローカルにまたは分散的に反応することができ、実施形態において、サービスとアクションの自管理に関わることができ。好ましい例示的な実施形態において、DMAの装置独立性は、例えばJAVAやJ2ME小フットプリントJAVA規格の使用によって拡張される。当然ながら、DMAは、この特定の実施形態に限定されず、すべての特徴を提供する様々な程度の複雑さと難しさを備えた任意の適切なソフトウェア構造に組み込むことができる。DMAのこの例示的な実施形態は、J2ME Connected Device ConfigurationをFoundation Profileと共に使用して、多数のリソースを有する大きなシステム構成要素から限られたリソースを有する小さいシステムまで、最も広い範囲の装置を使用可能にでき好都合である。この場合も、装置モデル・エージェントは、この実施形態に限定されず、装置の必要に応じて、JAVAやその他のプログラム言語の変形物において多くの他の実施形態が可能である。J2ME環境は、DMAソフトウェアが、装置および製品プラットフォーム全体にわたって装置に独立かつ再使用可能であることを保証することができる。J2MEは、また、ネットワーク接続された分散されたシステムのサポート、組み込まれたセキュリティ機能、およびコードの動的ダウンロードおよび操作のサポートを提供する。

#### 【0105】

実施形態は、例えばJAVAやJ2ME小フットプリントJAVA規格などのプラットフォームに依存しない規格の使用によるエージェントの装置独立性の拡張を含むことができる。当然ながら、エージェントは、そのような実施形態に限定されず、すべての特徴を提供する様々なレベルの複雑さと難しさを備えた任意の適切なソフトウェア構造に組み込むことができる。J2ME Connected Device ConfigurationをFoundation Profileと共に使用するエージェントの実施形態は、多数のリソースを有する大きなシステム構成要素から、限られたリソースを有する小さい組み込みシステムまで、広い範囲の装置を使用可能にすることができる。他の多くの実施形態は、エージェントが組み込まれるかまたはエージェントが提供する装置によって、必要に応じてJAVAやその他のプログラム言語の変形物を使用可能である。J2ME環境は、エージェント・ソフトウェアが実質的に装置独立でありかつ実質的に装置および製品プラットフォーム全体にわたって確実に再使用可能なままにできるので、好ましい環境である。さらに、J2MEは、ネットワーク接続された分散システムと、組み込みセキュリティ機能と、コードの動的ダウンロードおよび操作のサポートを含む。

#### 【0106】

以上説明した利点に加えて、DMAは、複数の異なるデータ・ソースを共通提供者APIの後ろに隠すことができる。これは、さらにソフトウェア・エージェントからの装置の詳細を抽象化する。実施形態において、個別の4つのデータ・ソースを、共通提供者およびCIMの後ろにひとまとめにすることができ、それによりサービスは、データ・ソースの詳細を知る必要がない。例えば、EPSV、PWS、CANバス、およびWeb UIにおいて、データをこの方法で管理することができる。また、装置モデル・エージェントに含まれる提供者層とCIMを、所定の製品または装置のために容易にカスタマイズできるようにする実施形態において、1組のツールを提供することができる。これにより、装置モデル・エージェントを採用または維持するプログラムが、データ・ソースへのCIM要素のマッピングだけを対象とし、装置モデル・エージェント全体の管理を対象とする必要がないので、再使用と速度のリリースが促進される。

#### 【0107】

システム構成要素のエンド・ツー・エンド・アキテクチャのイネーブリング機能は、

10

20

30

40

50

実施形態において、様々な分散構成要素の間で使用される通信方法の適切な抽象化を含む。この抽象化は、物理的接続機構ならびにセッション・レベルまでのプロトコルに適用されることが好ましい。両方のレベルにおけるそのような抽象化は、通信方法の詳細を分散構成要素から隠すのに役立ち、それにより、それらの構成要素は、サービスの動作に集中することができ、構成要素は、通信媒体またはプロトコルの変化から分離される。例えば、これにより、システムは、サービス自体がどれが使用されるかに注意することなく、無線リンク上で電子メールを使用したり専用のイーサネット・リンク上でウェブ・サービスを使用したりすることができる。

#### 【0108】

このタイプの抽象化は、装置にとっては新しいものであり、いくつかの重要な利点を提供する。それは、任意の所定のユーザに、システム構成要素の配置のフレキシビリティを提供することである。セッション層までの物理接続とプロトコル接続の任意の所定の組み合せから予想することができるサービス品質の情報を伝えることができる。システムは、様々な構成のサービス品質を監視して、特定のユーザに提供される所定のサービスが必要とするサービス品質を提供する通信リンクの有効性を評価する構成要素を、ホスト/バックオフィス側に備えることができる。これは、エンド・ツー・エンド・システム全体のなかの対応部分と自己監視部分の要素である。

#### 【0109】

通信の抽象化は、また、いくつかのフォルトトレランスを提供する。1つの接続機構が、なんらかの理由で停止した場合、通信モジュールが、それを検出し、失敗したその接続を、サービス品質が変化した可能性があるという事実以外をシステムの他の部分が知ることなしに、別の動作している接続と置き換えることができる。

#### 【0110】

実施形態において、代替として、サービスは、装置またはプロキシ内に「ハードコード」されてもよい。すなわち、サービスの追加および削除と関連した多数の管理機能が必要ない。装置で動作するサービスの組み込み部分は、分散構成要素間のウェブ・サービス・トランザクションに従わなければならない。これにより、バックオフィスは、システム内の完全な動的サービスと同じように「ハードコード」されたサービスを有効に処理することができる。

#### 【0111】

ハードコードされたサービスは、バックオフィス加入によって使用可能にることができる。これにより、サービス提供者は、任意の所定の装置上で使用可能にされた特定のサービスを制御することができ、これは、サービス提供者に、提供されたサービスが業務ニーズに基づいてどのように市場に広まるかを決定するフレキシビリティを提供する。例えば、サービスは、パッケージの一部でもよく、無料で提供されてよく、有料で提供されてもよく、更新を必要としてもよく、完全なサービス提供のために別のトランザクションを必要とするトライアル・ベースで提供されてもよい。

#### 【0112】

ハードコードされたサービスは、特定の要件のための共通の基本的挙動と一定の拡張機能を共用することが好ましい。サービスは、一緒に動作するが、インテリジェント・プロキシ上および/またはバックオフィス・サーバ内の組み込みプラットフォームにおいて装置自体上で動作する構成要素を有することが好ましい。これらのサービスは、ハードコードされるが、供給業者/サービス提供者のバックオフィス内のサービス・ライフサイクル管理システムによって構成し管理することができる。

#### 【0113】

実施形態が使用することが好ましい規格のタイプには、分散管理タスクフォース（D M T F）、ウェブ・ベース企業管理（Web Based Enterprise Management）（W B E M）、および共通情報モデル（C I M）がある。前に説明したように、C I Mは、実施形態に、サービスの再使用を可能にする装置モデルおよび抽象化を提供する。さらに、実施形態は、ウェブ・サービス、X M L、様々なバージョンのH T T P、およびS S Lを使用する。

10

20

30

40

50

また、実施形態は、例えばVeriSignからのサーバ側証明書を使用することができ、この証明書は、ファイアウォールとインターネットを横切る通信を使用可能にする。装置およびバックオフィスにおけるアプリケーション環境を使用可能にするために、実施形態は、例えば、Java 2 Micro Edition (J 2 M E)、Insignia CorporationによるEmbedded Virtual Machine、Java 2 Enterprise Edition (J 2 E E)、BEA WebLogic 7.0 Application Server Technology Suite、およびOracle8iを使用することができる。当然ながら、これらは、単なる例であり、必要に応じて他の構成要素を使用することができる。さらに、現在予測されておらず実施形態のシステムに追加することができる新しい構成要素が開発される可能性が高く、そのような構成要素は、実施形態の範囲内にある。提供されるサービス、そのライフサイクル、および特定の製品のD M T F C I Mの拡張機能は、実施形態の範囲内にある新しい技術の例である。

10

#### 【0 1 1 4】

実施形態は、さらに、既に配置済みのシステムへの新しいサービスの迅速な追加および配置を可能にする。例えば、新製品の発売後すぐに、現地で最初の3ヶ月の動作から学んだ教訓に基づいて新しい診断サービスが開発される。このサービスの正確な性質と挙動は、製品が発売されたときは予想できず、したがって診断サービスは、発売された製品には含まれていなかった。実施形態により、そのような診断サービスを、導入された装置に実質的にいつでも追加することができる。

#### 【0 1 1 5】

実施形態は、新しい「サービス・バンドル」が含まなければならないサービス・モデルと内部仕様を意図している。したがって、サービスの権限と構成情報の他に現地の既存の装置に新しい機能を追加しなければならない場合に、新しいコードをダウンロードすることができる。この機能は、実施形態に従って、新しい機能を容易に受け入れるように設計された装置に組み込まれたサービス・プラットフォームと共に使用することができる。さらに、実施形態の組み込みサービス・プラットフォームと共に使用されるとき、D M Aに組み込まれたC I Mによって装置独立の抽象化が提供されるので、プラットフォーム全体に新しいサービスの新しいコードを再使用することができる。そのようなプラットフォームのない装置では、新しいコードを、例えば現地でパッチを当てるかアップグレードするためのさらに特殊なソフトウェア・ダウンロード・サービスとしてさらに追加することができるが、そのようなサービスを使用可能にするコードは、ほとんどプラットフォーム固有であり、したがってあまり再使用できない。

20

30

#### 【0 1 1 6】

実施形態のこのシステムは、診断ルーチンや他のサービスを、装置のプラットフォームにとってきわめてフレキシブルな方法で提供する。そのような使用可能にされた装置は、バックオフィスのサービス提供者にとって、実施形態による他の各D M A対応装置と同じように見える。さらに、装置内部のプラットフォーム上でローカルに動作する装置系列のサービスはすべて、インテリジェント・プロキシを介してではなく供給業者システムに直接通信することができる。

40

#### 【0 1 1 7】

配置のもう1つの変形は、D M Aを製品自体に完全に組み込むことである。この実施態様は、それらが両方ともD M A対応プラットフォームであるという点で、例1の実施態様ときわめて類似している。しかしながら、この例では、小 footprint D M A プラットフォームは、製品に組み込まれ、印刷機構インターフェース・プラットフォーム (Print Station Interface Platform) (P S I P) と組み込み型装置コントローラの両方と通信する。その製品は、小 footprint プリント・システムによって必要とされる限られたリソースを受け入れ可能であり、必要なインターフェース構成要素の配置と一体化は、比較的容易である。

#### 【0 1 1 8】

再利用可能なD M Aは、既にJ V Mを有するシステムへの「臨時収容所 (drop-in)」である。小 footprint D M Aは、システム・リソースを消費するものではなく、その

50

のようなプラットフォームの実施を大幅に高速化することができる。

#### 【0119】

##### 「自動メータ読取り」

実施形態に組み込まれた配置フレキシビリティを使用するもう1つの例は、エンド・ツー・エンド・サービスの観点からシステムを見ることによって理解される。この場合、サービスは、自動メータ読取り(automated meter reads)である。このサービスは、ループ内に入れることなく、一般に電話、ファックス、電子メールまたはウェブ・エントリによって受け取った毎月または4半期ごとのメータ読取りを取得することを重視している。これは、読み取りの精度と適時性の両方を高め、ユーザの時間を節約し、供給業者に納品書発行と請求書発行を改善させることができる。

10

#### 【0120】

装置から要求されるデータが、小さくまた大部分が既に使用可能であるため、インテリジェント・プロキシを使用することができ、すべてのSNMP対応製品の関与を容易にすることができる。これは、DMA対応であるが完全にはSNMPに準拠していない装置と共に使用され、すなわち、広い有効範囲を簡単に達成できる。この場合も、この事例での抽象化とシステム・モジュール性が重要である。バックオフィス・システムは、装置が供給業者とどのように接觸したか（直接またはプロキシを介した）を知る必要がなく、必要なものは装置の通し番号だけであり、予定されたときにメータ読取りを要求することができる。装置のサービスへの関与を可能にする方法と、バックオフィスのサービス提供者から要求を出す方法をこのように分離することは、配置フレキシビリティを提供する際に利点である。

20

#### 【0121】

##### 「早期警報システム」

実施形態において、現地でのより多くの試験をサポートするために、レポート・システムやリモート監視サービスやその他のリモート・サービスが組み合わされて、1組のツールが構成される。基本的なシステムとデータ収集サービスは、人間の観察と報告に依存するデータ収集システムを補うことができる。同時に、システムの組み合せは、設計チームが製品の問題解決活動を基本とすることができるより大きく統合された知識を提供する。さらに、現地の装置から収集したデータの共通モデルは、プラットフォームをまたがって使用することができる報告ツールおよび基本パフォーマンス報告機能を配置する機構を作成する。

30

#### 【0122】

##### 「リモート制御および装置サービスによるプレミアム・リモート・アシスタンス」

実施形態の基本原理の1つは、装置自体が自分のライフサイクルとサポートに積極的に関わらなければならないことである。これは、トラブルまたは状態を報告するいくつかの状況において有効である。これは、現地のシステムが良好に動作し続けるようにするために、システム性能を監視しソフトウェアまたは構成を自動的に変更することができる装置に組み込まれた診断エージェントと共に動作することもできる。しかしながら、ユーザが経験する多くの問題は、装置の故障と関連したものと同じくらいユーザの問題と操作エラーに関連したものがある。さらに、周知のように、記録装置は、複雑な電気機械システムであるため、必ずしも遠隔で修復できるとは限らない。

40

#### 【0123】

現地の装置の操作サポートのニーズに対応し、現場のオペレータと共に作業する新しい方法を支援するために、この実施形態では、リモートUIと人間対人間のサポート・システムが組み合わされている。サポート自動化ソリューションは、高品質なサービスとサポート提供物の一部分を補うことができる。これは、データ収集とリモート監視を自動化し、また前述の多くのリモート・サービスを提供する。また、この組み合せにより、追加の訓練、問題解決およびソフトウェアの微調整が必要とされるときに、共用UIによって装置のオペレータと一緒に直接作業してオペレータを支援する優れた方法が提供される。

#### 【0124】

50

「接続トレードオフ」

図4-1に、ユーザ・サイトの装置とバックオフィスの間の通信リンクのいくつかの例示な選択肢を示す。A、BおよびCと示した3つの主な選択肢がある。選択肢AとCだけが、装置と供給業者のバックオフィス間の接続を独立で完成させることに注意されたい。選択肢Bは、供給業者に戻るリンクを完成させるために、AまたはCに接続する必要がある。

【0125】

各選択肢の長所と短所のまとめを表2に示す。

【0126】

【表2】

10

20

30

40

	有線接続	無線接続	
	選択肢A :	選択肢B : ローカル	選択肢C : 長距離
長所	<ul style="list-style-type: none"> <li>インターネット・アクセスにより現場のすべてのネットワーク装置に可能。</li> <li>LANとインターネットを使用するためのユーザまたは供給業者のコストの追加がない。</li> <li>プリンタ／独立型オプションに関係なく電話線を追加することができる。</li> </ul>	<ul style="list-style-type: none"> <li>最終的にプリンタで無線接続が使用可能になる。</li> </ul>	<ul style="list-style-type: none"> <li>セルラ・オプションと双方向ページャの両方のオプションの技術がある。</li> <li>いくつかの供給業者から既製品として購入したり、使用可能な標準的な技術から供給業者用に特別に最適化したりすることができる。</li> <li>供給業者の既存のページャ／セルラ・サービス・プランを活用する場合は、未接続の装置にとってよい結果になる可能性がある。</li> <li>この選択肢は、未接続の装置にとって、コストが回されず、ユーザ・フィードバックが有効であった。</li> </ul>
短所	<ul style="list-style-type: none"> <li>ユーザ・ネットワーク上のトラフィックが若干増える。</li> <li>未接続のシステムがLANオプションによってカバーされない。</li> <li>使用可能にし動作させる代償として電話線オプションが追加される。</li> <li>電話線接続の維持が難しいことが分かった。</li> </ul>	<ul style="list-style-type: none"> <li>未接続のプリンタでは無線機能がまだ使用できない。</li> <li>ネットワークの他の部分への接続用の無線アクセス・ポイントが存在しない場合がある。</li> <li>供給業者へのリンクを完成させるためには、ローカル無線がさらにAかCのどちらかを必要とする。</li> <li>接続される装置が増えると無線ニーズが一時的なものになる場合がある。</li> </ul>	<ul style="list-style-type: none"> <li>双方向ページャと携帯電話アドオン構成要素のコストが高い。</li> <li>ユーザが、別のネットワークに無線でも接続されているネットワーク接続装置のセキュリティを心配する。</li> <li>有線接続に対して限られたバンド幅。</li> <li>接続される装置が増えると無線ニーズが一時的なものになる場合がある。</li> <li>アドオン・ボックスを各タイプのプリンタに接続するために、アドオン・ボックスが、独特な通信インターフェースを必要とする可能性が高い。</li> </ul>

表2

【0127】

接続選択肢はすべて、異なる手段で供給業者に入る場合でも、同じバックオフィス・インフラストラクチャを再使用することが好ましい。

【0128】

選択肢は、グループとしては、様々なユーザ要件を満たす配置のためのフレキシビリテ

50

イを高めることができるので、すべての選択肢が魅力的である。好ましい接続方法は、実現可能なとき、選択肢AのLANとインターネットを介した有線接続である。これは、最小の開発投資と最小の作業費用の選択肢である。短期間ではこれが特に有力であるが、サービスの価値が検証されており、リソースは、初期のサービス開発に焦点を合わせることが必要とされ、装置に接続する追加の方法を提供しない。しかしながら、この選択肢だけを追求すると、初期にサービスから外された未接続の装置には対応されない。当面の間、各サービスは、提供物に未接続の装置を手作業で含める方法を検討しなければならない。

#### 【0129】

次に好ましい接続方法は、選択肢Cの携帯電話または双方向ペーディヤの技術による長距離無線である。システムは、この構成で、有線装置によってシームレスに動作することができ、この機能を使用可能にすると、何かユーザ問題が発生したときにその問題を解決できるようになる。しかしながら、多数の製品の全体にわたって大規模に無線機能を配置した状態ではいくつかの問題がある。例えば、現地のきわめて幅広い多数の製品と互換性を持つようにいくつかの異なるアドオン・モジュールを開発することは、ほとんどのシステムが、詳しい装置データおよび動作にアクセスする同じインターフェースを持たないため、コストが高くなる可能性がある。さらに、無線接続と通信コストを追加する追加の費用は法外に高価なことがあり、その接続を使用していくつかのサービスが利用できるようになる。単純でより容易に配置可能な無線構成には、提供できるサービスの数とタイプに固有の制限があり、そのためコストの正当化が難しい。最終的に、無線接続が、供給業者／サービス提供者または別の関係者がファイアウォールを回避してそのネットワーク上にある他のリソースにアクセスする可能性がある方法であるため、ユーザは、無線接続も有するネットワーク接続システムに対する心配を表明した。

#### 【0130】

最後に、選択肢Bは、ローカル無線接続である。この方法は、ローカル無線接続技術が、我々のユーザ環境およびプリンタに具体的にどのように組み込まれるかにより使用することができる。

#### 【0131】

「装置サービスのサポート・エンド・ツー・エンド・インフラストラクチャ」  
ユーザ・サイトの装置をレガシ・システムと業務プロセスに接続するためのエンド・ツー・エンド・インフラストラクチャをサポートすることが必要とされる。図1と図2に示したエンド・ツー・エンド・システムは、エンド・ツー・エンド・インフラストラクチャの初期の例示的な実施形態である。これは、基本的なデュアル・モードの装置の関与（直接およびサービス・プロキシ経由）をサポートし、初期のサービス通信と加入モデルを使用し、例えば供給業者の環境でホストされたエッジ・サーバを介してサービス・データおよびアクションの共通エントリ・ポイントを使用する。エッジ・ホストは、追加の実施形態に適した形で区分することができるが、1つのシステム上に物理的にホストされてもよく、それにより、市場獲得と採用が増大したときに準備コストが最小になる。

#### 【0132】

10

20

30

【表3】

イネーブリング技術の進化	インフラストラクチャの進化
<ul style="list-style-type: none"> <li>ハイエンド装置は、ユーザが選択した場合に環境内で見つけた他の装置のサービス・プロキシとして働くことができる。</li> <li>装置は、直接またはプロキシを介してサービスの組合せに関与することができる。</li> <li>装置が供給業者のいずれかによって開始されるセキュアな通信を、高レベルのサービス保証を必要とするサービスに配置することができる。</li> <li>CWWは、SNMPだけに対してCIMプロトコルを介して新しい装置と通信することができる。</li> <li>装置は、供給業者への直接の無線接続によってサービスに関与することができる。</li> </ul>	<ul style="list-style-type: none"> <li>エッジ・ホストは、サービス対応とトランザクション管理に集中する。</li> <li>供給業者は、すべての装置からのCIMデータが記憶され、様々な内部供給業者機能からアクセス可能な共通DataMartを維持する。</li> <li>データをDataMartに送る他に、加入とサービス・ニーズに基づいてデータ/イベントがサービス・スパンサに直接送られる。</li> <li>特定のエッジ・ホスト、データベースおよびサービスの間の物理的分離。</li> <li>装置と通信の認証/許可を供給業者システムが集中的に処理する。</li> <li>新しいサービスを迅速に配置するために、関与する装置をどのように構成しなければならないかを規定するビジネス・ルールを定義するサービスの対応およびソフトウェア/ツールを確立し使用する。</li> </ul>

表3：技術とインフラストラクチャの例

## 【0133】

表3に列挙した各分野は、技術的配置の分野あるいはサードパーティのCOTSシステムを取得し調査しなければならない分野を表す。これらの分野は、また、技術の十分な要件がまだ分かっていない分野を表す。

## 【0134】

前述のように、ユーザ支援自助プログラム(user assisted self-help program)、装置中心サービス(device-centric services)、および/またはECATのリモート監視をサポートするように元々設計されてない印刷製品は、そのような提供物が、初期出荷の高速化と製品の連続した成功に重要なことがある。そのような製品のニーズは、その状態に関してまたユーザがどのように使用してきたかに関して現地の装置から報告を毎日(または何らかの他の期間)受け取ることである。我々は、このサービスをリモート監視と呼んできた。これは、プログラム・チームが、現地で問題をより早期に識別できるようにし、販売、マーケティング、およびサポートがその出力も改善する重要な情報を提供する。

## 【0135】

これに対する1つのソリューションは、コントローラ上に、コントローラからIOTにローカルに接続された装置中心サービス(Device-Centric Services)(DCS)の装置モデル・エージェント(DMA)120を提供することである。アドオン構成要素またはカスタマ・サポート・プラットフォーム(Customer Services Platform)(CSプラットフォーム)115は、このニーズに対するソリューションである。CSプラットフォーム115は、いくつかの既存のインターフェースのうちの1つまたは複数を介してIOTにローカルに接続し、データおよび機能の見方を統一し、ローカルの機能の操作および管理をローカルUIに提供し、サービス・プラットフォーム124およびAPIにリモート接続および装置中心サービスを提供する組み込みシステムの形を取ることができる。CSプラットフォーム115は、装置中心サービスのフレームワーク内のサービス・プラットフォーム124によって使用可能にされるDMA120と組み込みサービス層141の両方の製品の実施形態である。

## 【0136】

図12、図16～図19および図21を参照して、CSプラットフォーム115は、ネットワーク接続された組み込み型パーソナル・コンピュータの形をとることができることが好ましい。さらに、アドオン構成要素は、ヘッドレス・ボックス(headless box)の形

をとることができる。どの特定の形においても、アドオン構成要素 115 は、少なくとも 1 つの物理的インターフェースを介して I O T に接続される。C S プラットフォーム 115 の U I は、ローカル・ネットワーク上の任意のブラウザで使用でき、C S プラットフォーム 115 の組み込みウェブ・サーバ 130 によって提供される。好ましい実施形態において、ユーザは、C S プラットフォーム 115 のローカル U I として、自分の D F E 上のブラウザを使用する。C S プラットフォーム 115 は、ローカル・ネットワーク・プロキシ、ファイアウォール・パスワード、D N S サーバ I P アドレスなどを知るために、任意のブラウザが構成されるかのように、ネットワーク接続され構成されることが好ましく、それにより、インターネット 400 上で使用可能なエッジ・サーバ 410 に接続することができる。動作するとき、C S プラットフォーム 115 は、この接続を使用して、メッセージと命令を検査し、加入されたサービス 140 をサポートするために必要なデータも送る。エッジ・サーバ 410 は、装置サービスのエンド・ツー・エンド動作と関連した待ち行列、メッセージ、サービス、およびトランザクションを管理する。

## 【0137】

C S プラットフォーム 115 は、マザーボード 701 と、L i n u x などの組み込みソフトウェア・オペレーティング・システム 704 を備えた低コストの組み込みパーソナル・コンピュータを利用したプラットフォームであることが好ましいが、他のオペレーティング・システムを使用することができる。アドオン構成要素 115 は、補助入出力やスタティック・メモリ・ボード 702 などのハードウェアによってカスタマイズすることができるが、そのようなカスタマイズは、コストを抑えるために最小であることが好ましい。構成要素 115 は、内部ハードウェア・プラットフォームが時間の経過と共に変化して、プラットフォームのコストを 2 / 3 に削減することができる最小汎用パーソナル・コンピュータの価格曲線に従うことができるように設計されている。内部記憶媒体 703 として、ハード・ディスク・ドライブ以上に信頼性を高めたコンパクト・フラッシュ（登録商標）・メモリなどのメモリを使用することができる。また、コンパクト・フラッシュ・メモリを使用すると、配置された新しいサービス 140 に追加の記憶リソースが必要な場合に、C S プラットフォーム 115 をアップグレードするコストがドガり、さらに、コンパクト・フラッシュ・メモリは、システムにとって通常のハード・ディスク・ドライブと同じように見える。さらに、アドオン構成要素 115 に標準的なパーソナル・コンピュータ技術を使用すると、費用曲線に従う迅速な改訂が可能になり、標準的なアドオン技術（例えば、ウェブ・カメラ）がプラットフォームと互換性をもつことが容易になりかつ保証される。

## 【0138】

C S プラットフォーム・アドオン構成要素 115 と装置の I O T の間の接続パスの例には、Fuji Xerox プロトコルおよびインターフェースの Electronic Partner (EPSV) 712 ~ 714、R S 422 および／または R S 232シリアル・ポート 715、716、サービス・ラップトップを装置に接続するためにカスタマ・サポート・エンジニアによって使用される接続 P W S 717 ~ 718、C A N バス接続 719 ~ 721、および U S B （図示せず）がある。また、デジタル・フロント・エンドに対する独自のインターフェースのような追加のインターフェースを監視して、サービスおよびシステム管理活動のための追加のデータを提供することができる。特に、C S プラットフォーム 115 は、そのような接続に限定されないように設計されることが好ましいため、他の接続も実施形態の範囲内にある。

## 【0139】

ルータ 730 を含み、ルータ 730 は、別の接続がアクティブになる場合に多数の情報を管理し、かついくつかの活動のプリエンプションを操作する役割を持つことが好ましい。したがって、通信は、通信がデータ破損問題なしに行われれるかのように実現される。

## 【0140】

組み込みソフトウェア・システムは、前述の診断ルーチンなどのローカルにホストされ

10

20

30

40

50

た機能と、動的に追加し構成することができるサービスの両方をサポートしてフレキシブルな構成要素を提供することが好ましい。したがって、実施形態は、装置中心サービス・プラットフォームに基づいて DMA120 を組み込んだシステム構成要素と、CS プラットフォーム 115 がシステムのローカル・イネプラとして働いて装置中心サービスに積極的に関与できるようにする組み込み J VM およびウェブ・サーバとを意図している。

#### 【0141】

組み込み DMA120 は、デジタル・フロント・エンドおよび／または DMA120 を独力で実行する能力に関係なく、サービス 140 を装置 110 から直接提供できるようになる。これにより、装置 110 は、DCS サービス・モデルを介してサービス提供物に積極的に関与することができる。アドオン構成要素 115 は、また、システムのまわりに構成される新しいサービス 140 のプログラマチック・インターフェースを提供し、それにより、製品と一体化された迅速かつ頑強なソリューションが可能になる。さらに、アドオン構成要素 115 内にウェブ・サーバ 130 を含むため、IOT のインターフェースとしての CS プラットフォーム 115 とリモート・サービス提供物の間での直接ウェブ・サービス・トランザクションおよびサービスが可能である。

#### 【0142】

カスタマ・サポート技術者ではなく訓練されたユーザに合わせてカスタマイズされた IOT 診断ルーチン 740 は、既に IOT によって提供されている既定の診断ルーチンのための使い易くかつ国際化された UI を提供する。診断ルーチンは、例えば、トナー密度レベルを最適化し、均一な画像品質を得ることができる（最高設定 741）。提供することができる他のサービス 742 には、Belt Edge Learn があり、これは、横方向の位置決めとベルト操作性能を改善するために新しい中間ベルトの縁に関して学習するルーチンである。Belt Edge Learn の目的は、2つのベルト・エッジ・センサを使用してベルトの動きを追跡することである。これらのセンサから受け取ったデータを使用し、IOT は、ベルト・トラッキング・ローラ／モータ／センサを使用して自動的に調整して、ベルトが内側寄り／外側寄りの動きなく回転するようになる。さらに他のサービスには、例えば、画像位置決めシステム上に IOTにおいて見つかった完全な画像をセットアップするセットアップ・ルーチンである RegiCon と、システムによって印刷されるハーフトーン密度を調整するセットアップ・ルーチンである Half-tone がある。ハーフトーン・パターンの印刷により、ページ全体にわたって一定の階調レベルがユーザ設定可能になる。ほとんどすべての画像品質欠陥がハーフトーン・パターンで現れるので、ハーフトーン・パターン自体は、問題を診断するために使用される。

#### 【0143】

CS プラットフォーム・アドオン 115 は、組み込みウェブ・サーバを介してウェブ・ベース UI を使用することが好ましい。これは、CS プラットフォーム 115 自体のハードウェア・コストを削減し、代わりに、ほとんど必ず存在しかつネットワーク接続された装置 110 のデジタル・フロント・エンドと関連付けられたモニタ、キーボードおよびマウス・ハードウェアを使用する。また、これは、ローカル・ネットワーク上の適切なブラウザによって、他のネットワーク接続された PC からアクセス可能である。そのような UI は、時間の経過と共に新しいサービスおよび機能の高い操作性と拡張性を提供する。このアプリケーションだけの GUI を提供するコストは、法外に高くなる。実施形態において、ウェブ・ベースの UI は、文脈依存ヘルプとコール・センタや他のサポート・サイトへのリンクを含むことができ、システムをより使い易くする。UI は、例えば、ハードワイヤード・ネットワーク接続されたパーソナル・コンピュータを含むユーザの LAN 上で接続された任意のブラウザで使用可能である。さらに、無線アクセス・ポイントが、CS プラットフォーム 115 に直接接続されているかまたはユーザのネットワーク上に取り付けられている場合に、互換ブラウザを有する無線接続されたパーソナル・コンピュータまたは手持ち式コンピュータを UI として使用することができる。

#### 【0144】

実施形態の装置中心サービス・アドオン構成要素 115 は、事前にロードされ許可され

10

20

30

40

50

たいいくつかのサービスを有することが好ましいが、これは不要な場合もある。C S プラットフォームは、加入したサービス提供物の装置中心サービス・モデルに従う。P D T は、基本のサービス・セットをイネーブルすることを決定した。さらに、構成要素は、リモート・サイトからの通知により、C S プラットフォーム・ファームウェアのソフトウェア・ダウンロードとリモート・アップグレードのためにイネーブルされることが好ましい。

#### 【0145】

システムは、同期サービスによって、リモートD C S ホスト310または410と定期的に連絡して、新しいトランザクションが待っているかどうかを確認することが好ましい。そのうちの1つは、新しいソフトウェアがシステムに使用可能であるという可能性がある。使用可能な場合は、管理タブから使用可能なアップグレード・ステータス画面によってユーザに通知することができる。ユーザは、また、ソフトウェア・アップグレード画面のリフレッシュ・ステータス・ボタンによって手動で最新版を確認するように選択することができる。アップグレードが可能な場合、ユーザは、そのアップグレードを受け入れる選択をすることができる。受け入れた場合、ソフトウェア・ダウンロード・プロセスは、自動的に、必要な最新版をダウンロードし、それをインストールし、古い方のバージョンを保存し、システムを再起動する。

#### 【0146】

構成要素115は、e Service 提供物をサポートする供給業者にセキュアな暗号化された通信を戻す。実施形態の診断ルーチンは、動作が完全にローカルな場合でも、サービスとして扱われる。次に、そのサービスを、すべてのサービスに使用されるサービス加入モデルによって制御することができる。これにより、必要になったときに、C S プラットフォームの機能を有效地に無効にすることができる。

#### 【0147】

システムは、初期のサービス・セットをユーザに提供できることが好ましい。そのような初期の提供物には、例えば、自動請求書発行、自動供給品補充、リモート監視がある。自動請求書発行は、要望に応じてまたは自動的に、装置中心サービスのインフラストラクチャを介して、必要な伝票発行メータを供給業者に報告する加入サービスであることが好ましい。自動供給品補充は、名前が示す通り、メータ供給品を人間の介在なしにユーザのサイトに適時かつ正確に送出するようにするために必要な情報を供給業者に提供するために、トナー使用量イベント、領域有効範囲イベント、およびトナー・ボトル変更イベントを追跡する加入サービスであることが好ましい。リモート監視は、定期的に、システムで見つかった構成可能なデータ・セットを収集し、それを標準的な方法でモデル化し、それを供給業者に戻すサービスであることが好ましい。そのサービスにおいて見つかったデータのタイプの例には、請求書発行メータ、I O T 故障、媒体経路詰まり、画像領域有効範囲、媒体使用量（重量、サイズ、およびタイプ）、機能使用量、トナー・ステータス、片面／両面印刷量、媒体トレイ使用量、縮小と拡大、コピー・モード、および高頻度サービス項目のステータスがある。

#### 【0148】

適切なシステム動作を保証するために、システムに追加のサービス・セットを組み込むことができる。例えば、D M A ハウスキーピング・サービス、健康モニタ、D M A - I O T 間通信ステータス・モニタ、サービス同期サービスなどであり、D M A システムのリモート部分と定期的に連絡をとって、D M A が行うべき新しい指示または活動があるかどうかを確認するサービスである。

#### 【0149】

セキュリティを保証するために、アドオン構成要素115は、実施形態において標準的なセキュアなウェブ・データ伝送技術および証明書を使用する。例えば、V e r i S i g n 証明書、R S A 暗号化、S S L、および関連した技術を使用することができる。さらに、アドオン構成要素115は、ユーザが装置から送られたすべてのメッセージを調べることを可能にする詳細なトランザクション・ログを提供することができる。C S プラットフォーム115から送られたトランザクションは、すべて送信または暗号化のためにパッケ

10

20

30

40

50

ージされる前に、XML形式でログをとることができる。これは、本当に言っていることだけを送っているという供給業者の記述の信頼性を高めるため、ユーザによる別の検査機能層を提供する。

#### 【0150】

実施形態において、データをエッジ・ホスト410に送る前に3つの許可レベルを呼び出すことができる。契約上の取り決めは、データが自動的に送られ、またユーザが伝送ログを調べることができることを述べていると期待される。様々な取り決めをすることが必要なユーザのために、複数の許可レベルに対応するオプションが、システムに組み込まれている。レベルは、トランザクション・ログにすべてのトランザクションの記録を維持する監査(Audit)およびログ(Log)と、バックオフィスへの送信が達成されたときに画面メッセージ、電子メールまたは何らかの他の手段によってユーザ代表者に通知する単純通知(Simple Notification)と、バックオフィス300へのメッセージの待ち行列を維持し、待ち行列が空でないときにユーザ代表者に通知する承認後送信(Approval Before Sending)とを含むことができる。承認後送信において、ユーザ代表者は、必要に応じてメッセージを調べ、次にデータの送信を認めることができる。デフォルト許可レベルは設定可能であるが、好ましい出荷デフォルト・レベルは、監査とログである。事前支援自助ツールおよびIOT自体の診断アクセスでも、すべての機能に1つのパスワードを有する。以前のシステムには、複数の人々の役割に対応し、それに従ってパスワードを管理する方法はなかった。

#### 【0151】

実施形態において、イネーブルされる多数の役割には、例えば、技術主オペレータ(Technical Key Operator)(TKO)、カスタマ・サポート技術者(Customer Service Engineer)(CSE)、およびシステム管理者(System Administrator)(SA)がある。任意の所定の役割のアクセスを構成するシステムは、ウェブ・ベースのGUIによって提供される。パスワードは、最初、それぞれの役割に個別化された共通のパスワードに設定されることが好ましい。実施形態のシステムは、SAが、自分のパスワードを構成しTKOのパスワードを管理できることを意図しており、これにより、標準的なIT産業のプロセス、プロトコル、および手順を使用して、ネットワーク接続された役割ベースのパスワード管理が可能になり、かつ任意またはすべての役割に関してリモートで認証されたログインおよびパスワード管理が可能になる。リモート・ログインは、訪れるどのCSプラットフォーム115上でも同じパスワードを使用したいCSEには特に魅力的なことがある。リモート・ログインの認証は、パスワードのみ、パスワードとトークンの組み合わせ、または他の任意の適切な方法でよい。これは、リモート・ホスト・サイトに対するCSプラットフォームのネットワーク接続性によって制限され、バックアップ(または、ローカル)共通CSEまたはユーザの役割パスワードが、提供される必要がある。

#### 【0152】

プラットフォームは、忘れたローカル・パスワードをリモートでリセットするプロセスを含むこともできる。SAは、ヘルプ・デスクを呼び出し、話している本人として首尾良く認証される。ヘルプ・デスクは、CSプラットフォーム(IOT通し番号で示された)に命令を出して、それをそのSAパスワードをリセットさせる。SAは、SYNCボタンを手で押して、CSプラットフォームにエッジ・ホスト410と連絡をとらせ、SAパスワードをリセットする命令を受け取り、操作を完了するように伝える。他のすべてがCSに失敗した場合は、プラットフォーム・ファクトリ・リセット手順に従って、すべてのパスワードをデフォルト構成にリセットすることができる。

#### 【0153】

通常のDCSサービス加入および活動プロセスによって、新しいソフトウェア・サービス140を、CSプラットフォーム・アドオン構成要素に追加することができる。加入了サービスは、DMA120とDCSエンド・ツー・エンド・システム1によって自動的に管理し導入することができる。これにより、CSプラットフォーム115は、時間の経過と共に新しいサービスを提供することができる。新しいソフトウェア・アップグレード

10

20

30

40

50

は、CS プラットフォームのリモート・ソフトウェア・アップグレード機能によって提供することができる。これにより、CS プラットフォーム 115 のより重要なアップグレードを、技術担当者がユーザ・サイトを訪れることなく、ユーザ承認を受けて実行することができる。これにより、コストが大幅に削減されるので、システム・アップグレードを配置することができる頻度が増える。

#### 【0154】

以上の特徴ならびにほとんどのシステム・ハードウェアおよびソフトウェアへのCOTS 技術の使用のために、プラットフォームにリモートで追加された適切なサービスによって新しいハードウェアを追加することができる。ハードウェアの拡張を必要とする新しいサービスの例は、ユーザにとってウェブ・カメラを利用したサポートになる。低コストのUSB ウェブ・カムを追加することにより、抱えている問題のスナップ写真を撮ってその写真をヘルプ・デスクまたはコール・センタに送ることができるので、CS プラットフォーム 115 は、ユーザが電話上よりも適切なリモート・サービスを受けることができるサービスを加入ユーザに提供することができる。

#### 【0155】

実施形態は、CS プラットフォーム 115 と同じサブネット上でネットワーク接続されたパーソナル・コンピュータ上のCS プラットフォームの導入を意図している。インストール・プロセスは、その概略図を図19に示し、標準的なネットワーク・ユーティリティと、CS プラットフォームの後ろにあるLED 表示の組み合わせを使用して、インストーラのプロセスを進める。CS プラットフォーム 115 は、ヘッドレスな組み込みシステムであることが好ましいので、インストール・プロセスは、注意を要する場合がある。ここに示したステップは、インストールを行う1つの可能な方法であるが、他の方法も可能である。コマンド画面上のフィードバックと装置のLED の組み合わせが、インストールのための確実なプロセスを提供する。構成要素 115 は、最初、電源投入待機状態であり（ブロック 801）、ユーザによって電源投入される（ブロック 802）。ステータス LED または類似のものが、構成要素 115 が起動中であることを示すために点滅し、次に、構成要素 115 が準備完了状態になったときに点灯することが好ましい（ブロック 803）。実施形態において、ユーザは、構成要素 115 のMAC アドレスを読み出し（ブロック 804）、UI でコマンド・ウインドウを開き（ブロック 805）、MAC アドレスやその他の情報と共にコマンドを入力する（ブロック 806）。次に、ユーザは、構成要素 115 をピング（ping）して（ブロック 807）その構成要素をテストし、次に、1つまたは複数のLED が点灯状態になるような完了の表示を待つ（ブロック 808）。次に、ユーザは、ブラウザで構成要素のウェブ・サーバ 130 まで進み（ブロック 809）、管理者としてログオンし（ブロック 810）、必要に応じてネットワーク情報を構成して（ブロック 811）、構成要素 115 がエッジ・ホスト 410 と通信できるようにする。構成要素 115 が再起動し、その間に、IOT の電源を遮断しなければならない（ブロック 813）。両方の再起動が完了した後、インストールとセットアップが完了する（ブロック 813）。

#### 【0156】

固定IP アドレスの使用およびIP アドレスを取得するDHCP の使用を含む、ネットワーク接続のための複数の方法でCS プラットフォームを構成することができる。固定アドレスは、ほとんどのユーザにとって好ましく、準備ができたときにブラウザをCS プラットフォーム UI に示しやすくするという利点を有する。代わりに、DHCP は導入がきわめて容易であるが、CS プラットフォームとDNS サービス接続に装置ドメイン・ネームを必要とする。自動ドメイン・ネームを提供する1つの可能な方法は、IOT 通し番号を、MAC アドレスの最後の2桁と組み合わせることである。ユーザに既知の容易に使用可能な情報とデフォルトのCS プラットフォームの他の組み合わせが可能である。

#### 【0157】

CS プラットフォームは、任意のブラウザが構成されるのと同じように、ネットワーク用に構成される。これは、CS プラットフォーム UI 上で書式に記入することによって手

作業で行うことができる。また、ウェブ・ブラウザ・プラットフォームに既にある設定を確認するルックアップ機能をOSが備えている場合は、その機能によって行うことができる。これは、基準となる設定値を提供し、次に、ユーザは、必要に応じてそれをカスタマイズしたり修正したりすることができます。構成した後で、ユーザがネットワーク管理ページを終了する前に、供給業者エッジ・サーバ410とすぐに連絡をとって設定が正しいことを確認しようとするテスト構成ボタンを提供することができる。

#### 【0158】

CSプラットフォーム115のユーザは、CSプラットフォーム・ウェブ・ページへのブックマークをなくす場合があり、ユーザがそのウェブ・ページを再び見つけ易くする方法を提供する必要がある。システムを構成するためにDHCPが使用される場合は、ユーザは、指示に従うだけでCSプラットフォームのデフォルトまたはハードコードされたドメイン・ネームを決定することができる。また、ユーザの環境においてDFEまたはネットワーク接続されたパーソナル・コンピュータ上にインストールされ動作し、またすべてのCSプラットフォームを見つけ表示する発見ツールを提供することができる。この発見ツールは、供給業者のウェブ・サイトからダウンロードすることもできる。ツールへのリンクは、CSプラットフォームUIから使用することができ、したがって、CSプラットフォームIPアドレスの紛失に備えてダウンロードし保存することができる。また、このツールは、インストール中にDFEにローカルに保存するオプションによって、CSプラットフォーム上にローカルに記憶させることができる。

#### 【0159】

前述のように、ルータは、CSプラットフォームへの並列アクセスの方法を管理する。CSプラットフォーム・ルータは、供給業者ゲートウェイおよびDMA要件ならびに現地の装置110と互換性をもつことが好ましい。ルータは、ローカルPWSポートとIOT診断（シリアル）ポートの間で直接接続を提供する。実施形態のルータは、また、例えばIOT診断（シリアル）ポートを介してネットワーク・クライアントにネットワーク接続を提供し、IOT CANバスならびに様々な装置110のEPサービスへのネットワーク経路をサポートすることができ、すべての（EP以外）通信トラフィックおよび優先順位を調停する。優先順位は、動作モードが滑らかに移行できるようにイネーブルされる。例えば、DCUソフトウェア・アップグレードのアプリケーション・セッションの優先順位、IOTシリアル・ポート3のローカルPWSポートIOT診断セッションのもう1つの優先順位、および他のネットワーク・セッションのもう1つの優先順位がある。通常のシステム動作では、「オープン」なローカルPWSセッションはプリエンプションされないことが好ましく、ローカルPWSセッション要求は、ネットワーク診断アプリケーション・セッションを中断できることが好ましい。すべての中止が適切でなければならない。ネットワークDCUソフトウェア・アップグレード・セッションのリエンプションは、許容できないことが好ましいが、EPおよび/またはCANバス・セッションは、CANセッションが1度に1つだけ許容される限り、いつでも許容されることが好ましい。DMA

EPゲートウェイ・サーバ/クライアント・クライアントは、EPポートへの排他的アクセスを有することが好ましく、DCUV2.0は、ネットワーク上のローカルPWSトランザクションの経路をサポートすることが好ましい。

#### 【0160】

DCU v2.0のサポートにおいて、コミュニケーション・コントローラは、例えば、START\_DIAG\_SESSIONイベント（このイベントの顧客は、DMAプッシュ・イベント提供者である）を生成し、END\_DIAG\_SESSIONイベント（このイベントの顧客はDMAプッシュ・イベント提供者である）を生成し、DMAプッシュ・イベント・提供者へのPWS\_MESSAGE\_EVENTとしてLOCAL\_PWS\_PORTとIOT\_SERIALコミュニケーション・ポートの間で各メッセージを送ることによって、ネットワーク上のローカルCSE診断セッションの内容を提供することができる。

#### 【0161】

DCU v2.0のさらに他のサポートにおいて、コミュニケーション・コントローラ

10

20

30

40

50

は、例えば C T S (Clear To Send) 信号として指定された R S 2 3 2 信号を利用するこ  
とによって、システムに接続された P W S に存在を伝える方法を提供することによって、  
ローカル P W S ポートを介した I O T シリアル・ポートへのローカル C S E 診断接続をサ  
ポートすることができる。 C T S 信号は、 D C U に論理ハイ・レベルで保持されることが  
好ましい。

#### 【0162】

R S 2 3 2 インタフェースの D S R (データ信号レディ) 信号は、 I O T の診断モード  
を制御することができる。 D C U は、この制御を保存するか必要に応じて「作成」する。  
I O T が、診断モードで、電源が遮断された場合は、 I O T への D S R 信号がハイに設定  
されたときに診断モードで立ち上がる。 P S W は、 D S R 信号を制御する。

10

#### 【0163】

特定の実施形態を説明したが、出願人または当業者には、現在予測されずまたは予測す  
ることのできない代替、修正、変形、改良および実質的等価物が現れる可能性がある。し  
たがって、出願され修正された併記の特許請求の範囲は、そのような代替、修正、変形、  
改良および実質的等価物をすべて包含するように意図されている。

#### 【図面の簡単な説明】

#### 【0164】

【図 1】実施形態のアーキテクチャ全体の概略図である。

【図 2】実施形態のアーキテクチャ全体のもう 1 つの概略図である。

【図 3】実施形態によるサービス申し込みおよび配置の方法の概略図である。

20

【図 4】実施形態による配置オプションの概略図である。

【図 5】実施形態による追加の配置オプションの概略図である。

【図 6】実施形態による追加の配置オプションの概略図である。

【図 7】実施形態による追加の配置オプションの概略図である。

【図 8】実施形態による追加の配置オプションの概略図である。

【図 9】実施形態による追加の配置オプションの概略図である。

【図 10】実施形態による装置モデル・エージェントのより詳細な概略図であり、実施形  
態による装置、装置モデル・エージェント、サービス・プロキシおよびサービス・ホスト  
間の対話のより詳細な概略図である。

【図 11】実施形態による装置、装置モデル・エージェント、サービス・プロキシ、お  
よびサービス・ホストの間の対話のより詳細な概略図である。

30

【図 12】実施形態による追加の配置オプションの概略図である。

【図 13】装置モデル・エージェントとその装置との関係をより強調した、実施形態によ  
る追加の配置オプションの概略図である。

【図 14】別の観点から装置モデル・エージェントとその装置に対する関係をより強調し  
た、実施形態による追加の配置オプションの概略図である。

【図 15】実施形態による装置モデル・エージェントのサービス・マネージャの動作方法  
の概略的フローチャートである。

【図 16】実施形態による C S プラットフォーム・アドオン構成要素の概略図である。

【図 17】実施形態による C S プラットフォーム・アドオン構成要素のもう 1 つの概略図  
である。

40

【図 18】実施形態による C S プラットフォーム・アドオン構成要素の無線配置スキーム  
の概略図である。

【図 19】実施形態による C S プラットフォーム・アドオン構成要素を設定する方法の概  
略図である。

【図 20】実施形態による対応サーバの概略図である。

【図 21】実施形態による C S プラットフォーム・アドオン構成要素のより概略的な図  
である。

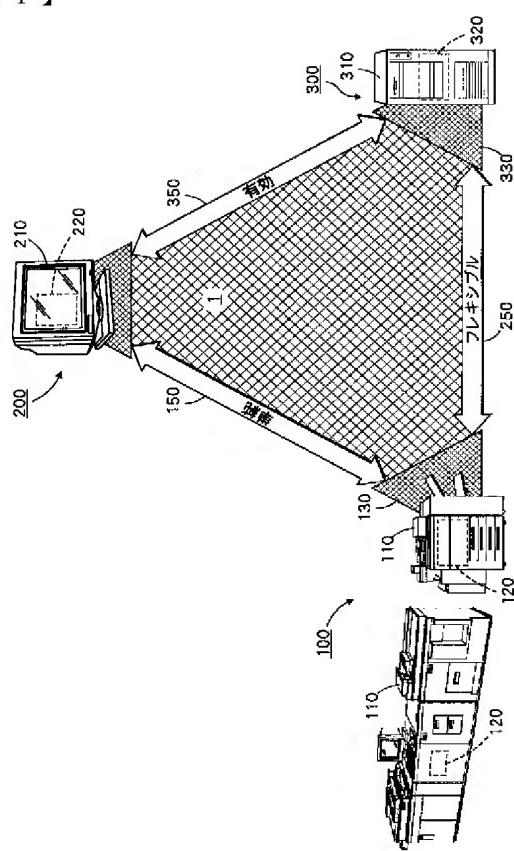
#### 【符号の説明】

#### 【0165】

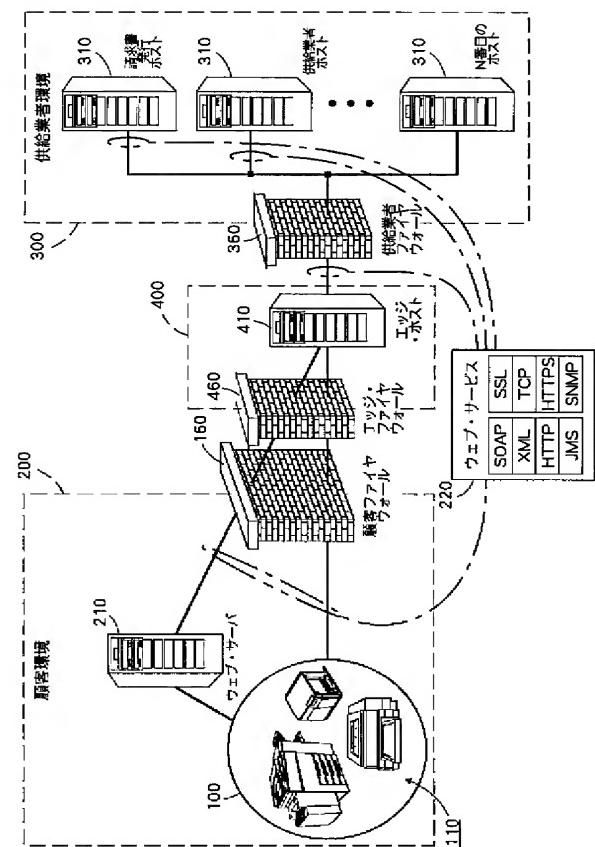
50

100 ユーザ環境、110 ローカル装置、120 分散装置エージェント、130 ウェブ・サーバ、200 アプリケーション・サーバ、210 装置プロキシ、220 ユーザ・アプリケーション、300 サービス供給システム、310 アプリケーション・サーバ、320 ホスト・システム。

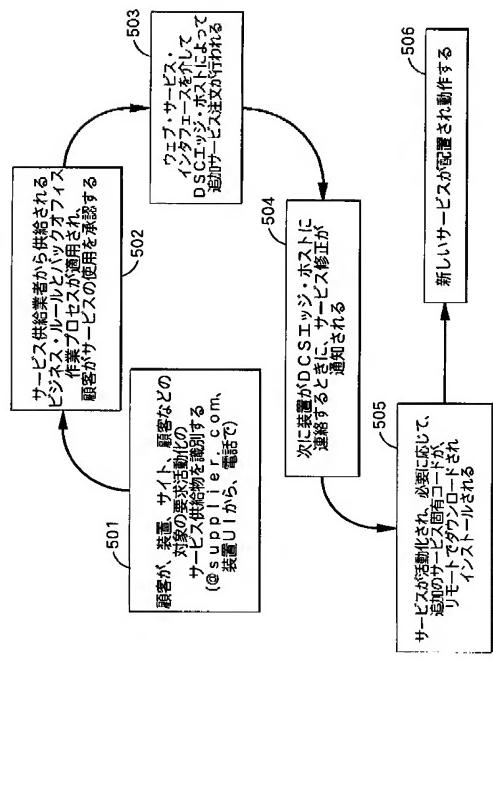
【図 1】



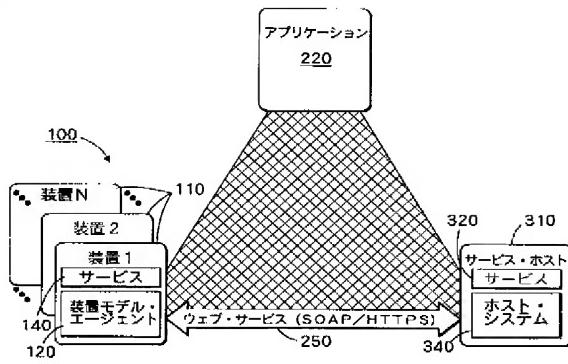
【図 2】



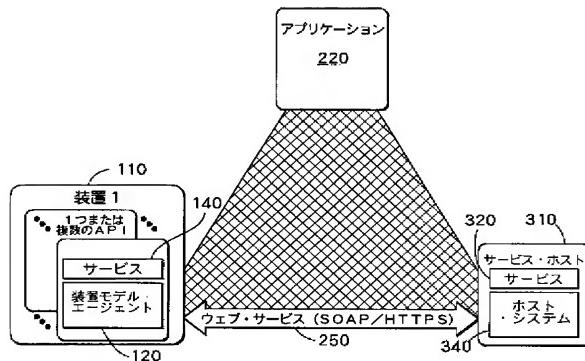
【図 3】



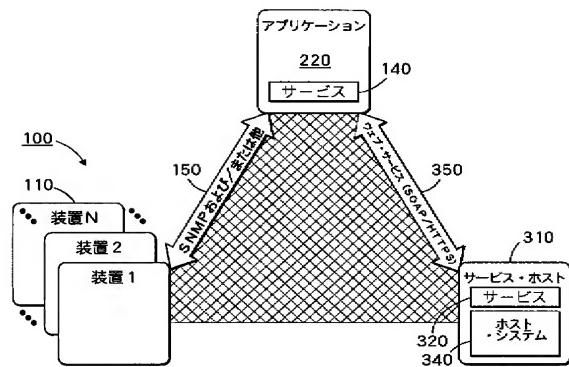
【図 4】



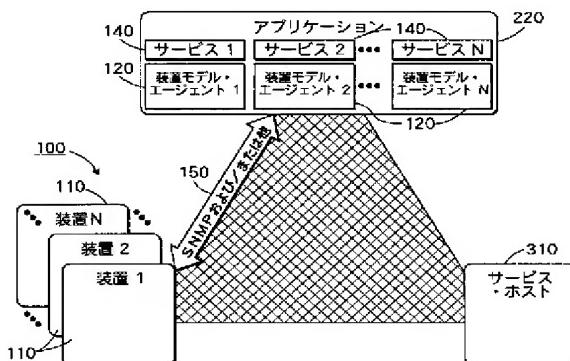
【図 5】



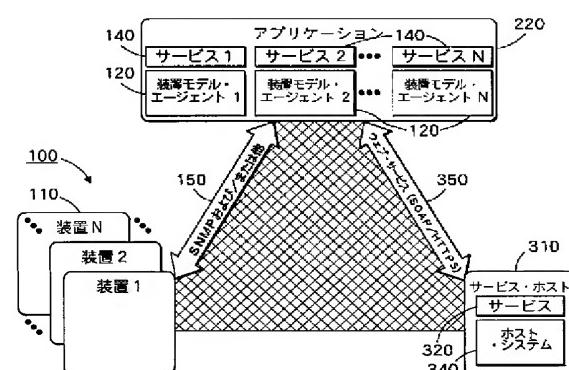
【図 6】



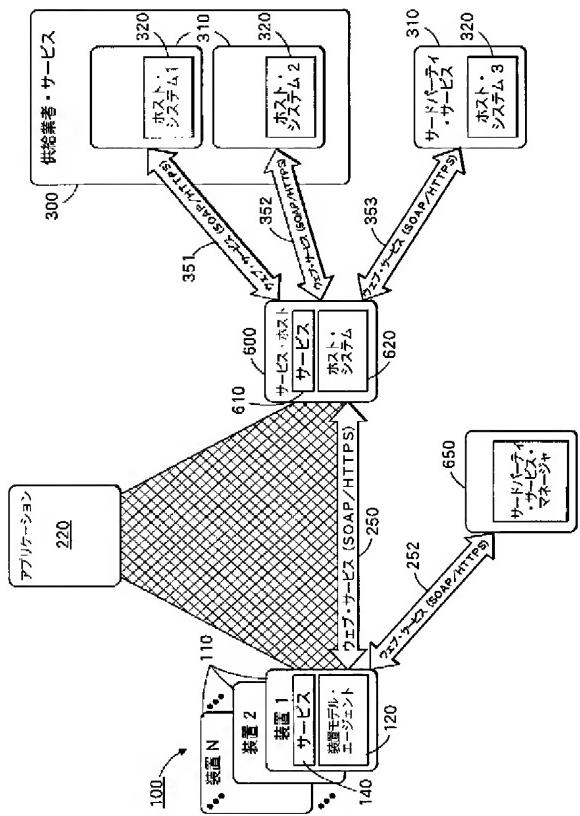
【図 8】



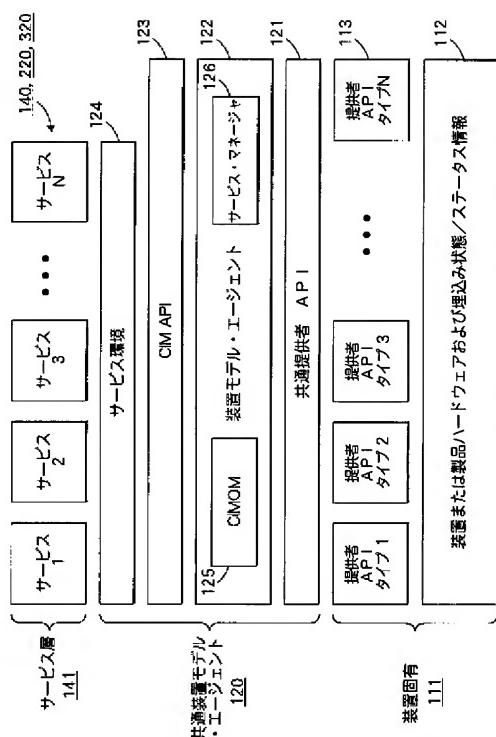
【図 7】



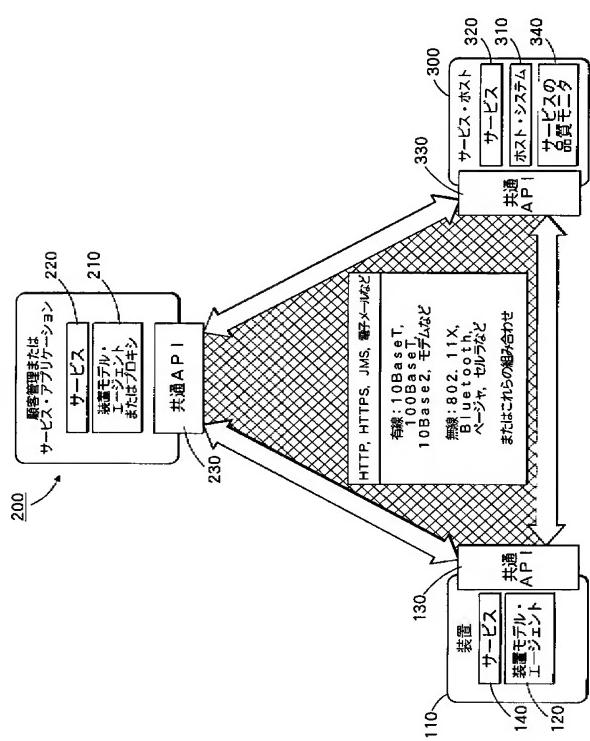
【図 9】



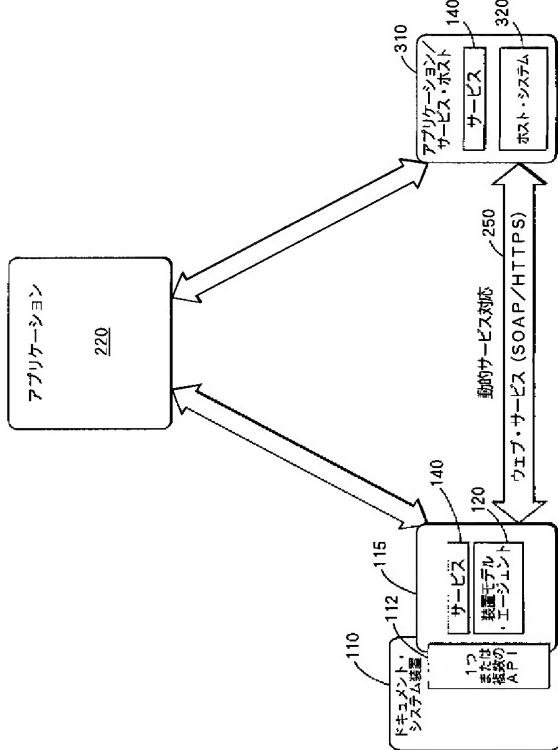
【図 10】



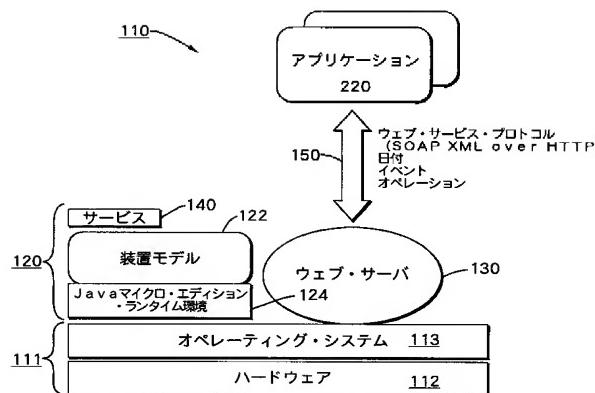
【図 11】



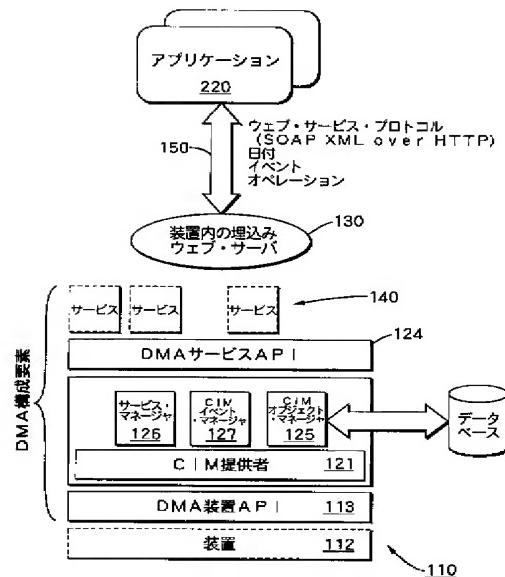
【図 12】



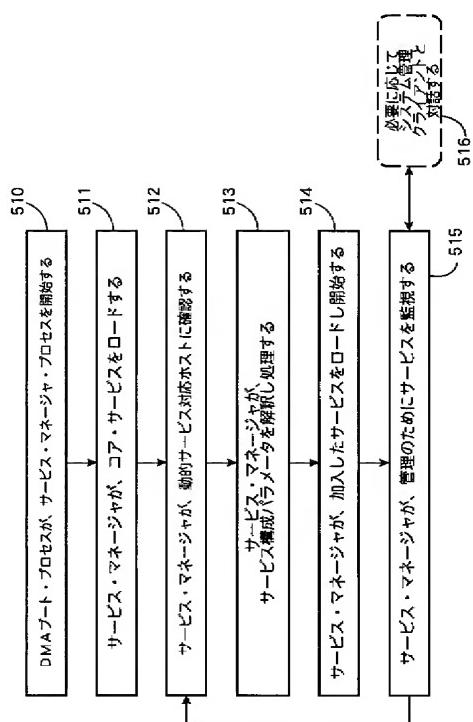
【図 1-3】



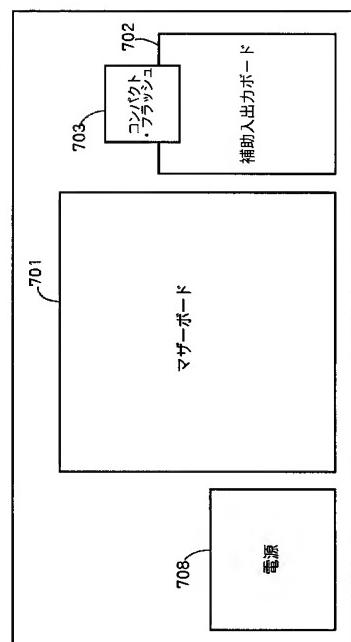
【図 1-4】



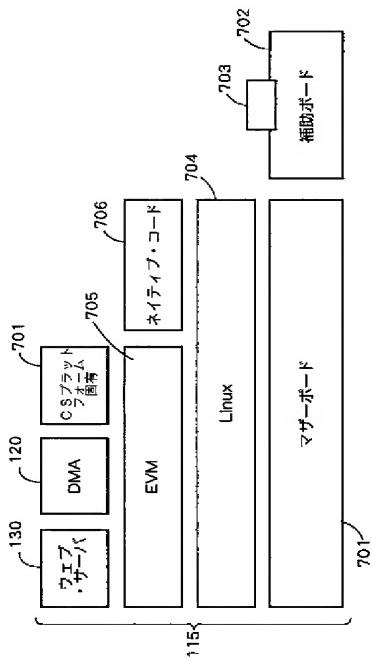
〔四〕 1 5 〕



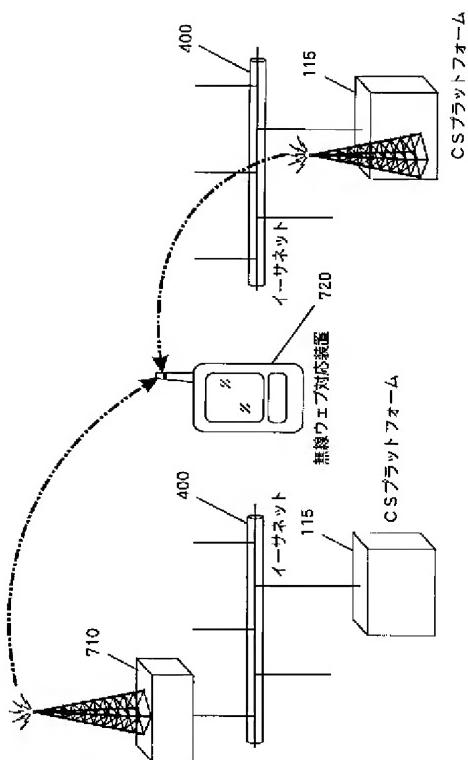
〔図 1 6 〕



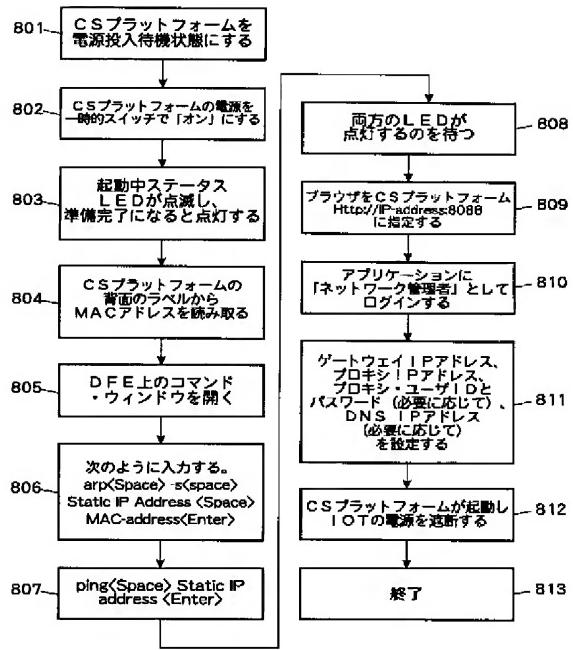
【図 17】



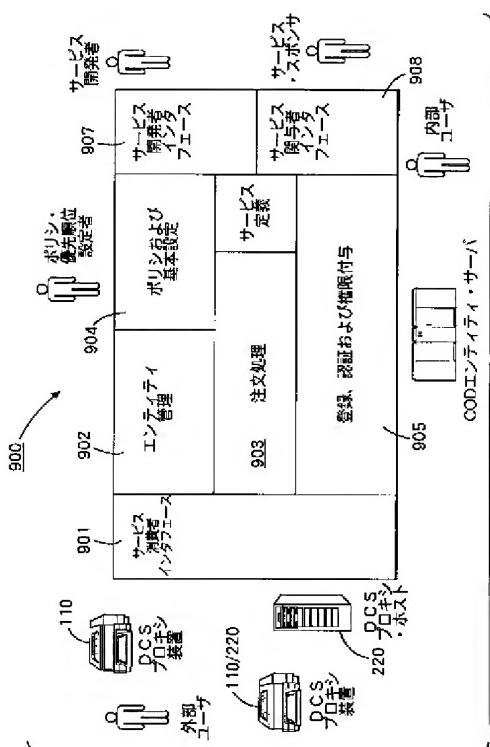
【図 18】



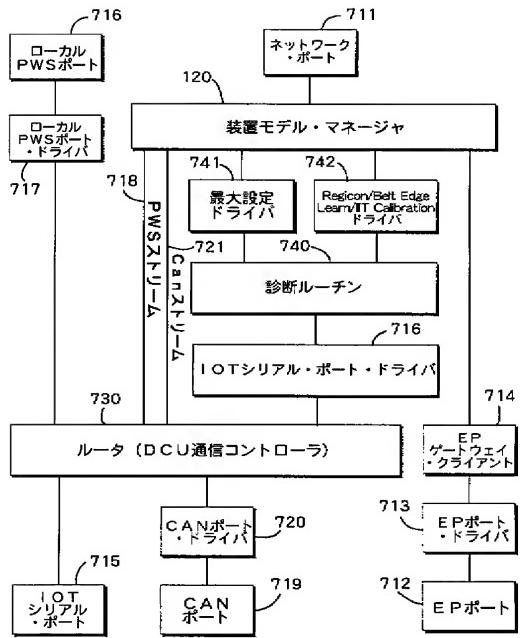
【図 19】



【図 20】



【図21】



---

フロントページの続き

(72)発明者 ロナルド エム ロックウェル  
アメリカ合衆国 ニューヨーク ロチェスター ウィルマー ストリート 67

(72)発明者 ネイヴィーン シャーマ  
アメリカ合衆国 ニューヨーク ペリントン コロニアル ドライブ 4

(72)発明者 クロード エス フィリオン  
アメリカ合衆国 ニューヨーク ロチェスター ローレルウッド ドライブ 97

(72)発明者 マイケル ピー ケホー  
アメリカ合衆国 ニューヨーク ロチェスター ロッキンガム ストリート 346

(72)発明者 アーチュロ エム ロレンツオ  
アメリカ合衆国 ニューヨーク フェアポート ヴァンダーバーグ ドライブ 8

(72)発明者 メアリー シー マッコーキンデイル  
アメリカ合衆国 ニューヨーク フェアポート ハルバート アベニュー 55

(72)発明者 ロバート ジェイ セント ジャックス  
アメリカ合衆国 ニューヨーク フェアポート マノーシャー ドライブ 104-3

(72)発明者 トレイシー イー シーレット  
アメリカ合衆国 ニューヨーク ウェブスター シャディ グレン サークル 608

(72)発明者 ジョン シー オースティン  
アメリカ合衆国 ニューヨーク ロチェスター ベレスフォード ロード 209

(72)発明者 マーク ディー ダニエルス  
アメリカ合衆国 ニューヨーク ピッツフォード ロック クリーク 120

(72)発明者 マイケル エフ カバノー  
アメリカ合衆国 ニューヨーク ウェブスター スティシィ レーン 10

(72)発明者 ウェイジア ヒュアン  
アメリカ合衆国 ニューヨーク ロチェスター クリッテンデン ブールバード 60 アパート  
メント 830

(72)発明者 クリストファー ジェイ レグルイット  
アメリカ合衆国 ニューヨーク ロチェスター アレクサンダー ストリート 397 コンドミ  
ニアム 16

(72)発明者 ロランツオ ウィットフィールド  
アメリカ合衆国 ニューヨーク フェアポート ヘブンウッド ホロー 48

F ターム(参考) 5B076 AB20 BB06 BB17  
5C062 AA02 AA05 AA14 AA29 AB38 AC41 AC42 AC56 BA04

(54)【発明の名称】装置モデル・エージェントとアドオン構成要素を含む、分散加入サービス、自動化供給物保守、  
および装置独立のサービスの実施を可能にする方法および装置、装置側の低コスト組み込みプラ  
ットフォーム、分散サービスの実施、および装置サービスの自主提供のための統合サーバ・プラ  
ットフォームのための方法および装置

【外国語明細書】

**METHOD AND APPARATUS FOR ENABLING DISTRIBUTED  
SUBSCRIPTION SERVICES, AUTOMATED SUPPLIES MAINTENANCE, AND  
DEVICE-INDEPENDENT SERVICE IMPLEMENTATION, INCLUDING  
DEVICE MODEL AGENT, ADD-ON COMPONENT, METHOD AND  
APPARATUS FOR LOW COST EMBEDDED PLATFORM FOR DEVICE-SIDE,  
DISTRIBUTED SERVICES ENABLEMENT, AND INTEGRATED SERVER  
PLATFORM FOR THE AUTONOMOUS PROVISIONING OF DEVICE  
SERVICES**

This application claims priority to U.S. Provisional Patent Applications Nos. 60/319,622 filed October 16, 2002; and 60/319,623; 60/319,624; 60/319,625, all filed October 17, 2002.

Embodiments relate to electroreprographic marking machines, facsimile machines, scanning devices, multifunction devices, and the like. Particularly, embodiments relate to the implementation and distribution of services such devices can offer users.

Installation of a marking machine or other business device is only the first step in the majority of its lifecycle. Most devices are involved in ongoing business processes between the product owners (users), the manufacturer of the product, and/or third party suppliers. Companies that manufacture marking devices typically include products and services in support of users' documents and hope the users will use and live with the offerings for quite a while. This post-sale period presents an opportunity for building a strong and mutually beneficial, long-term relationship between the manufacturer and the users. The post-sale relationship can be defined not only by what the devices do for users, but how they do it, how manufacturers support them, how manufacturers treats the users, and how easy it is to own and use the devices overall. Understanding this, embodiments addresses users' complementary needs to receive

services in support of the devices they use: post-sale lifecycles, break-fix needs, and integrated business processes are addressed in various embodiments. These processes range from break-fix service (repairs), to ongoing supply of consumables and supplies, to product upgrades, enhancements, and integration into solutions and other offerings. Traditionally, these post-sale processes were manual in nature and required the device owner/user to play an active role in relaying limited information to manufacturers and suppliers at the time of need.

Many electroreprographic marking machines, facsimile machines, scanning devices, multifunction devices, and the like provide services to assist with such processes that users must learn how to use or to avoid. Some devices also require meter reads and other types of maintenance that tend to irritate users. In the case of meter reads, users may have to read the meter on a monthly basis and communicate the results to a supplier via, for example, fax or phone. Additionally, users must manually check supplies of paper, toner, and other materials and place orders for new materials. Sometimes the number of services offered by a device can bewilder a user, leading the user to believe that the device is too complex to learn. Further, to avoid down time and other inconveniences, users often would rather make their own small repairs than call for a repair and wait for service to arrive and repair their device.

With the advent of modems, high-end products in user or user sites were connected back to manufacturers via phone lines changing this interaction model. The arrival of ubiquitous Internet connectivity and the proliferation of network connected products presents new opportunities to bring a more flexible and powerful approach to the integration of devices with post sale business processes. While network connectivity removes some of the drawbacks of phone line connectivity, systems described to date still carry many of the limitations that were associated with the interaction models developed for these early systems.

Disadvantages of current systems include tight coupling of communication method and system architecture, one-size fits all deployment and integration strategies,

and typically no support for devices already deployed. Systems that do offer support for devices already deployed typically are inconsistent between how already deployed devices and new devices are handled. Additionally, systems typically do not include an ability for rapid upgrade, extension, customization, and evolution of features, processes, and workflows and are often limited to basic business processes, failing to provide external services and solutions APIs in a consistent fashion. Generally, and almost across the board, systems treat the device as a simple repository of information, rather than an active participant in the services enabled. Devices must continue to have their mainline feature sets enhanced to stay competitive. In document systems, for example, speeds, feeds, image quality, and document workflows are typically characteristics that are enhanced to render devices competitive. However, increased post-sale interaction between devices, users, and suppliers, and the ability to integrate products into solutions and services and vice versa are becoming points of distinction between devices in the marketplace. In the near future, devices' success and value will likely be measured by the ability of devices to actively participate in their post-sale lifecycles, their ability to seamlessly integrate with solutions offerings, and their capacity for customization and extension based on user needs and requirements. The results of such device abilities are improved ease of use for the user, more effective support from manufacturers, and better overall user satisfaction.

A general industry trend for several years has been to take advantage of the increasing embedded computation and connectivity found in marking devices by offering remote services to increase user satisfaction and reduce operating expenses. This trend towards connected intelligent products started with remote services implementations on servers and other mission critical information technology (IT) related hardware and has become increasing prevalent in a variety of other industries, including marking devices. These remote services provide a win-win value proposition for both manufacturers and users. When implemented properly, these services allow for large cost reductions for the manufacturer, as well as a richer post sale experience for the user.

This transition will be driven by several coincident factors and needs. Competitive pressures and the need for improved internal business processes will require new ways of interacting with products in the field, as well as a shift in where responsibility for service and support resides. Manufacturers and users alike will prefer to be able to configure and add new features/services to products rapidly to solve immediate problems and to rapidly deploy new features. Simplifying and speeding this process will prolong the lives and enhance the value of deployed devices and will help keep users happy and productive. Manufacturers need to be able to provide these capabilities for new devices and those already deployed, but manufacturers cannot afford to be best in breed to everyone; devices must be able to easily incorporate third party or competitive elements. One size does not fit all, and multiple deployment configurations are necessary that give the manufacturer the ability to configure an appropriate solution for an individual user's needs. The manufacturer must also be able to make solutions behave consistently across multiple configurations so that they are manageable and supportable, and so the user remains in control.

Studies centered on determining user preference and need for these types of services conclusively point toward the need for new capabilities in offerings that will enhance the way users live with marking devices, billing systems, and supply chain. The studies also indicate that users desire these services and are willing to work with manufacturers to overcome security hurdles to implement them. In particular, the studies found that for nearly one third of users, these remote services would be likely to make users more loyal to a given machine brand at the time of next purchase. Most users would be willing to pay to acquire remote services capabilities on their machines are very or somewhat comfortable with sending data to services providers via the Internet, as long as they had some level of control over the data shared, and showed particular interest in directed self-repair, automated downloading of software, and remote supplies/services analyses and predictions.

In addition, an analysis of remote solutions state of the art shows that all major players in the marking device manufacture and remote solutions market offer some degree of remote service functionality and are placing increased emphasis on expanding these capabilities. In the offset printing market, integrating remote services into presses and peripherals is considered a cost of doing business.

Services offered to users prior to the instant system were assembled and managed end-to-end within specific product families. This required product teams to invest in developing, not only the product itself, but also the infrastructure, services, and back-office connections necessary to get the job done. This effort was often very difficult to sustain long-term and was often duplicated across product families.

Users' experiences can be greatly enhanced by simplifying the users' relationships with devices, such as, for example, marking devices. Embodiments can automate current, manually-performed and/or non-uniform business processes, as well as providing new workflows to address evolving user requirements. This will be accomplished by, for example, employing embodiments to enable devices to be active participants in their life cycles and value added services while keeping the users in control. Embodiments do this using standards architecture, such as Distributed Management Task Force and Common Information Model (CIM) based standards, to allow services to be written once for all devices employing and/or compatible with embodiments and to enable easy modular additions of new services on a product by product basis.

To achieve these ends, embodiments provide a common service model, services that work with a multitude of disparate devices, and flexibility in physical, logical, and operational configurations. Devices take on an active role in providing users with enhanced post sale experiences. Embodiments can enjoy seamless integration into back-office processes of both users and manufacturers.

More particularly, embodiments comprise a flexible end-to-end system for connecting devices to solutions offerings. Many deployment options in various physical

locations and configurations are possible to allow broadest device coverage and rapid deployment of capability for both machines in field and new products, while insulating device changes from back-office changes.

The system of embodiments can be reused across all compatible platforms, freeing individual platforms from the need to reinvent all back-office systems. Each platform team need only enable their product through one of the ways mentioned above and contemplated by embodiments, such as by embedding the DMA of embodiments and/or by complying with specific services transactions protocols.

An agent software component embedded into devices, add-on modules, and device proxies provides a common device model, common information management (CIM) application programming interface (API), and an environment in which device services can run. A common abstraction of a communication mechanism allows the system to be independent of the physical transport linking nodes. A service model supports services that run close to the device and their lifecycle, which includes the methods and processes for effective management and customization of services and solutions. As a result, services that are once written to the agent are capable of running on any device, add-on module, or proxy that includes the agent. This yields a system that enables devices and device proxies to be deployed and work together seamlessly from the point of view of the services, as well as policy-based provisioning for device-based services with both user and supplier inputs. The embedded service agent takes an active roll in solutions offerings and works in coordination with distributed solutions and/or a network-accessible server to provide required functionality. The server provides a clearing house for messages that must traverse the system and provides management functionality necessary to connect and customize distributed services at multiple levels of granularity.

In addition to increased user satisfaction and loyalty, embodiments can create financial benefits. Embodiments can provide cost savings from reduced service engineer usage through increased user self-help, remote diagnostics, and prognostics. In

embodiments including automated meter reads, reduced collection process infrastructure, better contract enforcement, and reduced reserves against inaccuracies can provide additional cost savings. Further, embodiments participating in automated supplies ordering can enable decreased inventories through increased accuracy of tracking consumables at user sites, in part due to more timely, accurate, and applicable measures. Additional cost savings could be realized in terms of eliminated phone time due to fewer call-in orders and disputes. Finally, embodiments can contribute to an increase in revenue from new services since so many users would be willing to pay a fee for the services offered by embodiments.

Embodiments respond to user need and interest by including, for example, a new class of remote services. These services will capitalize on the increased connectivity of devices in the user environment, and utilize embedded computations within the devices themselves to make devices active participants in simplifying user work processes. The platform enables a standards-based solution that can be used to modularly implement remote service offerings in a cross-platform manner that all use a common back-office integration and work processes. Specific examples of the types of services that can be offered in embodiments include: automated meter reads, automated supplies ordering, productivity reporting, software download, assisted user self-help, remote diagnostics, and prognostics.

Embodiments include a class of services that exist in support of the devices (printers, scanners, repositories, and even other services and solutions) and their lifecycles making them easier to own, use, support, purchase, and upgrade. Market research has shown that these services increase the value of devices to users and can potentially also increase their user satisfaction over the life of the product. This in turn should translate into higher user loyalty and consideration from our users when making new purchases.

These services, in embodiments, make use of new device capabilities including embedded device intelligence, take advantage of the increasing networked

population, and exploit information technology advances enabling devices to take a more active role in their post-sale life cycles enabling automated and expanded feature sets.

Embodiments provide the underlying set of components and their interconnections that enable suppliers to deliver these types of post sale services to users in an effective and efficient manner. The high-level goals defined for the platform have been used to drive the architecture and development of initial components and services. The detailed attributes of each support the four major goals for the platform. The major components of this system all work together behind the scenes to make the services offered behave seamlessly for users.

Embodiments provide for automated reporting of meter reads via phone, fax, or computer network. Additionally, embodiments automatically monitor supplies, warning users when supplies are low and allowing automated ordering of supplies then and in subsequent similar situations. Additionally, the services a device offers can be tailored to the users' particular needs, but can later be augmented or reduced as required by the user via automated service subscription, downloading, and installation offered by embodiments. Further, embodiments walk users through any operation they wish to perform, including small repairs and replacements of user replaceable units. An additional advantage of embodiments is the ability to manage assets of multiple devices from a central application.

For a general understanding of the present invention, reference is made to the drawings. In the drawings, like reference numerals have been used throughout to designate identical elements.

Embodiments provide a system 1 composed of several types of distributed software and hardware components that ensure physical and logical system design flexibility and responsibility of the components. Embodiments employ an architecture including, for example, devices 110 in the user/user environment 100, an asset management system 200 that can be in the user's network or environment 100, and a services host 310 that provides services 320 to which devices can subscribe. System management and services are provided in a system where devices are active participants in both their own services and lifecycle needs as well as those services and lifecycles in which they are only a part.

Using the Device Model Task Force (DMTF) Common Information Model (CIM) as a base, service management is added to the active behavior of a Common Device Model Agent (CDMA) 120. See particularly, for example, FIGS. 1 and 10. Each device 110 is preferably represented to the services host 310 by a CDMA 120 that communicates the status and configuration (part of 111) of its device 110, services 140 offered, and other information (additional parts of 111, for example) to the services host 310 using a common transaction language, such as DMTF CIM, for example. The DMA also provides a services environment 124 that is a runtime environment for services 140 on the device in which it resides, providing device-independence for the services offered by the services host 310. Thus, a particular service 140 can be written once and run on a wide variety of devices 110. This allows One to Many configurations, which can be enabled, required, and utilized as necessary. For example, embodiments can include device proxies, including simple proxies that include only behavior, and proxied device models, as will be expanded upon below. Also, embodiments can include devices that communicate directly with the services host.

The CDMA 120 includes the services environment 124, a CIM API 123, a CIMOM 125 and service manager 126 in the core DMA 122, and a common provider API 121. The common provider API 121 communicates with device-specific provider APIs 112 of the device 110 and retrieves information 111 about the device, including, for example, its configuration, status, and supplies levels. The common provider API 121 then makes such information available to the CIMOM 125 and service manager 126 in the core DMA 122, and services 140 running in the services environment 124. Thus, in embodiments, the DMA 120 can reside between a services layer 141 (the collection of running services 140) and device-specific APIs 112 and can communicate directly with the services host 310. In such a case, the DMA 120 must be connected to a communications medium, such as a telephone line or computer network, to enable communications with the services host 310.

Partly as a result of the use of the DMA 120, the system 1 in embodiments achieves substantial communication medium independence. Devices 110 can communicate with the services host 310 and or service proxy via network, land phone line, cellular communication, packet radio, pager based, Bluetooth™, IEEE 802.11, or any other suitable communications scheme. Such communications can be device initiated, host initiated, can be monitored and/or audited, and can use user preference, service offerings, and overall quality of service to determine which selections are appropriate for a particular scenario. The services 140 can be independent of device configuration; specifics of service content can be provided by the back-office or supplier 300, with service subscriptions being issued and validated by back-office business processes, allowing rapid introduction of new services.

Embodiments thus include an end-to-end system 1 assembled from many components using a unique combination of modularity, distributed computation, service models, and transactions. Embodiments employ an overall system architecture that supports rapid and flexible deployment of services in a modular fashion. Advantageously, this architecture employs abstraction of functionality and identification of system elements, common interfaces, and messaging models for communication between them. For device services, there are four main entities that can work together in a consistent and flexible manner: the devices themselves; management and service applications in the user's environment; back-office servers specializing in service management and service configuration; and business process integration servers and the business processes surrounding those services.

FIG. 1 shows a schematic representation of major components that can comprise the platform represented by embodiments. A small footprint embedded service platform 140 and intelligent agents 122 combined into the DMA 120 can be deployed and integrated with devices 110. Intelligent proxies for devices 110, enabling group management and participation in services, can be included in the platform, either as standalone applications or as part of other applications. In addition, such intelligent

proxies can enable legacy devices that are not themselves already enabled to interface with aspects of embodiments. A medium independent communications and applications infrastructure connected to a computer network or other communications network, such as the Internet and supplier Intranet, is employed that can securely and robustly connect fielded devices and products to the supplier and its internal systems. Additionally, embodiments include a set of industry standard web services technologies integrated with value-added extensions to enable those services. Embodiments thus provide set of services that run in and on an end-to-end system that support devices.

Devices 110 are typically physically located in user sites 100, though embodiments can accommodate devices 110 on other sites as well, and can be distributed around the world. A variety of devices 110 ranging from low-end products to high-end systems can be included. Embodiments use devices 110 that provide, for example, three main enablers in this system. Devices 110 provide a Common Device Interface (CDI) and a Common Information Model (CIM) 123 to enable easier integration with services by hiding device specific differences behind the interface. The CDI can be implemented in the CDMA 120 as the Common Provider API 121. This enables service reuse and greatly reduces the complexity of the system 1. The CDI is specified via a Distributed Management Task Force (DMTF) CIM with service-provider and/or manufacturer specific extensions for services and service management. The CIM 123 can also be based on the DMTF CIM with specific extensions better suited to devices with services, including diagnostics extensions, that can be provided by service-providers and/or manufacturers.

Devices 110 in embodiments of the inventive system 1 provide, for example, an embedded service platform 124 allowing services 140 to be delivered and run close to the device 110. The embedded service platform 140 also provides for local management of services, via service manager 126, and the ability to accept newly deployed services 140 asynchronously with software releases for the hosting platform. This reduces system complexity and speeds deployment of new or refined services to the field. Preferably, in

embodiments the DMA 120 provides the embedded services platform 124, though other systems could provide the platform. The DMA is preferably a software module that enables the embedded computational power, data, and functions of the device to be accessed and used in services that are deployed in a common fashion.

These components will typically be distributed across the user's environment 100 as well as at the supplier 300. Together, they provide a flexible end-to-end system 1 for connecting products (such as devices 110 and services 140) to post-sale solutions offerings (additional services 140). The system 1, in embodiments, is designed to provide an architecture in support of a series of deployment options in various physical locations and configurations. Preferably, embodiments provide the broadest device coverage and most rapid deployment of capability for machines in the field and new products in such a way that isolates changes at the device 110 from changes at the back-office 300. Embodiments further provide a unique, value-added, agent software component, the DMA 120, embedded into devices 110, add-on modules 115, and/or device proxies 210 that provide the common device model 122, DMTF CIM API 123, and new device services environment 124. Additionally, embodiments can provide a common abstraction of the communication mechanism(s) that allows the system to be independent of any physical transport linking the nodes (devices to supplier systems, etc.), providing greater flexibility and deployment customization based on user requirements. The service model of embodiments supports services that run "close to the device" and their lifecycle, which includes methods and processes for effective management and customization of services and solutions. Services in embodiments once written for the DMA 120 can run on any such enabled device 110 or proxy 220, and devices and device proxies can be deployed and work together seamlessly from the point of view of the services. Provisioning in embodiments can be accomplished on a policy basis for device based services based on both user and supplier supplied information, and services can be made available with rapidity.

The DMA 120 in embodiments takes an active roll in solutions offerings and works in coordination with the distributed solutions. These distributed device agents 120 work together with a server 310 at the supplier 300 accessible over a network, such as the Internet or a telephone system. The server's role is to provide a clearinghouse for messages that must traverse the solution and to provide management functionality necessary to connect and customize the distributed services at multiple levels of granularity.

For devices 110 already deployed that do not include this functionality, an option to add a physical system component 115 to the device 110, internally or externally, that enables this functionality is provided by embodiments. To the inventive system 1, a device 110 enabled in this fashion will look no different than a device 110 with the capabilities embedded, as long as the add-on component 115 has a rich interface to the device 110. For example, embodiments including such an add-on component 115 can have the component mounted on the input-output terminal (IOT) of a marking machine, connected to the IOT via EPSV, PWS, and potentially CAN Bus interfaces, and connected to a network. This configuration gives the IOT the capability to participate in device services 140. These add-on components 115 can then be found in a one to one mapping with the device because of the need to access non-standard, or non-network accessible APIs and interfaces in order to offer the full range of device capabilities to the DMA and services platform.

In order to accommodate the fact that the input output terminal (IOT) may come from another manufacturer and that a user can choose from many digital front ends (DFEs), embodiments feature an add-on system component. This add-on component can be, for example, a low-cost, embedded personal computer platform running an operating system, such as Linux or Darwin, and a Java Virtual Machine, such as, for example, Insignia's Jeode™ Embedded Virtual Machine, within which runs the DMA services platform. This add-on component can connect directly to the IOT through an interface, such as EPSV and PWS. The device is then also connected the user's internal network.

The devices management and service applications 220 in the user's environment back-office servers specializing in service management and service configuration business process integration servers and the business processes surrounding those services. The schematic representation of these system level components and their interconnections are shown in Figure 1.

Embodiments specifically relate to the role of devices in end-to-end system management and post sale application architecture and in offering services to users. The Device Model Agent (DMA) 120, the device side technology module in Device Centric Services (DCS) platform, is the main focus of this invention. DMA is a thin, efficient applications/services execution environment. The DMA provides a flexible, extensible, dynamic services management system allowing e-services to be designed, added, and managed within system without modifying the platform itself. While this invention specifically describes the integration of benefits from DMA into document system devices, the concepts are equally applicable in other domains. The DMA runtime environment is a thin software interface layer that resides on a document system device between the Java runtime environment and embedded web server.

The device model agent as described herein adds the following capabilities to document system devices. The unique combination of these capabilities enables several benefits related to system management application development, deployment, and maintenance.

The DMA 120 enables active participation in applications and services offerings, such as, for example, post-sale, system management, and other services. The devices 110 that embed DMA 120 can perform several computational tasks required in system management applications and services. In this architecture, an application server 200 installed, for example, in the user environment 100 or supplier 300 back office server 310, and the target device 110 collaborate to complete system management offerings. As shown, for example, in FIGS. 10, 13, and 14, the DMA 120 provides a service execution environment 124 where a service 140 may run as a whole or be part of a system

management application or host system 320 running on an application server or host system 310 of a services supplier 300.

The DMA services preferably can monitor device events and take prescribed actions. The DMA 120 can preferably publish data to subscribers/users upon occurrence of an event of interest and can preferably invoke methods, such as diagnostic routines, on the device 110 as directed by internal or external clients or users. This moves device specific processing closer to the device 110 from a centralized application server 320. The role of the applications server 320 transforms from a compute platform for execution of applications/services to the management and configuration of applications/services 140. Thus, devices 110 become active participants in the process, as opposed to being passive data repositories in strict client/server architectures.

The DMA 120 according to embodiments can also perform dynamic updates of services 140 and support components operating within the end-to-end DCS platform 1. Devices 110 that employ the DMA 120 can add new service components 140 dynamically. It allows a user or application component already on the device 110 to request such additions to support services 140. It can also allow the addition or deletion of components as needed and without system or DMA recompilation or restart. In embodiments, the target device 110 itself initiates the additions of a new or upgraded service as a whole or supporting components for existing services. Thus, in the system 1 described herein, the device 110 can now be responsible for initiating the activity to maintain itself and system management services running on it.

Embodiments additionally recognize the need for an application/services execution environment 124 to enable developers to work with consistent and standards-based tool set. The DMA 120 enables the development of device independent post-sale applications 140. Applications 140 written using DMA 120 interfaces do not have to change to accommodate new or upgraded DMA enabled devices. While the DMA 120 implements a model-based approach espoused by DMTF for achieving device independence for applications/services, it adds to this implementation a new component

called the service manager 126. The service manager 126 is primarily responsible for controlling service 140 lifecycle of each service 140 activated for the device. In addition, the service manager 126 preferably manages the services 140 and provides a programmatic interface (an API) for system management clients, local or remote, for control and management of services 140.

Operating within the end-to-end DCS platform 1, DMA-enabled devices and DCS application servers allow services provisioning and management by an application server or hosted systems 320 on a services host 310 of the supplier 300 or by a third-party service provider. The DMA-enabled devices 110 and application servers 320 collaborate to provide dynamic provisioning. Using this system, users can review a set of applications, select or customize one or more applications to fit their needs, and order the selected set. Subsequently, the applications can be installed, enabled, turned on, monitored, and/or managed.

In order to cover new and existing device base, the architecture of the DMA 120 in embodiments allows its deployment in several ways: For example, according to embodiments as shown, for example, in FIGS. 4, 9, and 11, the DMA 120 can be embedded in a networked device 110, such as a printer or multifunction device. In this embodiment, the DMA 120 becomes a web server side component. The DMA 120 can, for example, use Java Servlets, a standard method for hosting service-components behind a web server.

Alternatively, the DMA 120 can be embedded in a specialized hardware device or add-on component 115 to devices 110 that are standalone, such as copiers, or for existing devices in field that are not able to run the DMA 120. Such add-on components 115 are shown schematically in FIGS. 12, 16, and 17, and will be discussed in more detail below.

Another alternative configuration is for the DMA 120 to be embedded in a network application 220, either as a single device proxy or multiple device proxy

configurations. Such proxy configurations are illustrated, for example, in FIGS. 7, 8, and 11.

For any device manufacturer, post-sale applications can be important for continuing user loyalty. In case of the supplier of marking and/or multifunction devices, sale of a document system device is just the beginning of the user relationship. Continued service, support, supplies replenishment, and on-going maintenance become the main considerations that determine user retention rate. However, as mentioned earlier, the development, deployment, and management of post sale applications in a cost effective fashion itself poses several challenges. The complexity is driven by the presence of multiple stakeholders, including developers, device manufacturers, service owners, and customers/users. As a platform, the DMA 120 is preferably designed, according to embodiments, to bring all major stakeholders together and address the requirements of all stakeholders.

The DMA 120 preferably constructs a layer of abstraction between device specific interfaces 111 and system management applications. See, for example, FIGS. 10, 13, and 14. The abstraction provides a common view of device data, event, and operations to system management applications. The DMA 120 adopts a model-based approach to create device abstractions. The device models used for this purpose are based on industry standard efforts in DMTF (Distributed Management Task Force) consortium. An enhanced version of DMTF Common Information Model (CIM) is used as a basis. However, the implementation of common model with device interface is unique. Interactions between post-sale application clients and DMA are based in DMTF CIM Operations Over HTTP specification.

The DMA 120 enhances a DMTF/CIM implementation by adding a service manager component 126. The service manager 126 is responsible for loading services 140, maintaining a list of services 140 currently installed in the DMA 120, and management and lifecycle control of services 140. The service manager 126 preferably works as an automated process and can automatically look up and start services 140 as a

standalone component that can be accessed programmatically or through a DMTF CIM API 123. The service manager 126 can provide access to active services 140 on the device 110, as well as management data for one or more active services 140.

The DMA service manager 126 preferably supports core services that are started automatically when the service manager 126 loads. Such core services preferably do not require provisioning support. The service manager 126 also preferably supports subscribed services 140 that require enablement through a dynamic provisioning feature of the DMA 120.

FIG. 15 is a schematic flow diagram that illustrates the service manager 126 startup and normal execution. When the DMA 120 boots, it starts the service manager (block 510). The service manager 126 then loads the core services (block 511) and checks with the dynamic services provisioning host (block 512). The service manager 126 further interprets and processes service configuration parameters (block 513) and loads and starts subscribed services 140 (block 514). The service manager 126 then monitors the services 140 for management purposes (block 515) and interacts with a system management client as desired (block 516).

The DMA 120 is preferably written using a substantially platform-independent language, such as, for example, Sun's Java 2 Micro Edition (J2ME). The DMA is thus highly portable and can be used as a system component in any system configuration where Java is available.

Management and services applications 200, according to embodiments, can be located in the user's environment. Some applications 220 can enable the offering of device services by, in part, behaving as a services proxy for devices 110 that are networked but not fully enabled to participate actively in device services by themselves as shown schematically in FIG. 7, for example. In other words, the applications 220 can act as proxies for networked devices that do not have the DMA 120 or the software to support the service offerings 140 directly. For example, some applications, such as Xerox® Corporation's CentreWare Web® (CWW), can act as device proxies for Simple

Network Management Protocol (SNMP) enabled devices. To the extent that the SNMP agents in the devices deliver the required data and functionality to support the services being offered, this can be a good way to bring devices into the services fold. In such a case, the application, such as CWW, assumes the responsibility for hosting the services 140 that run close to the device 110.

To the back-office or hosted portion 300 of the end-to-end system 1, such as the services host 310, the device 110 looks nearly the same when proxied via applications 220 as it would if the device communicated with the services host 310 directly. The difference between proxied devices and direct devices is substantially no different than the differences associated with direct devices with different levels of firmware. The device capability variations can be managed on the back-end/services host 310 via a provisioning system for device-based services. The impact of these variations is that advanced services utilizing very specific capabilities of a given device become less portable; that is, services written for product specific sections of the CIM extensions will not be completely portable and may not be as compatible with other devices. However, services written to the Core and Common models will remain portable, and deployment issues will be managed by the provisioning system.

The applications 220 can also behave as services proxies for devices that are not networked directly, but have an add-on connectivity option. Such a connectivity option can be hard wired, such as Ethernet, or wireless, such as Bluetooth™ or IEEE 802.11, and can be local or more expansive in its coverage. For example, a stand-alone copier with no network connection for printing can have a small wireless LAN connection has been added, such as an 802.11b or other wireless network. The proxy behaves in all the same ways as when a networked device without DMA is proxied, but the proxy now also includes the hardware required for the wireless access point used to communicate with the devices to which add-on connectivity has been attached. An example of such a system would be CWW installed on a server that is both physically networked on a LAN and also has a wireless access point attached. To the back-office

edge host 310, the devices 110 proxied in a wireless fashion look no different than those proxied on the LAN.

The applications 220 can also enable consolidated management of services from a server in the user environment. Device proxies can provide, in embodiments, an aggregation and group management function for services associated with their proxied devices. This can, for example, be a graphical user interface (GUI) for a system administrator or user to look at the status of services on a set of devices.

The common device interface of services and their transactions to the back-office systems, such as services hosts 310, can be an extension of the interface used on the devices themselves. This enables the services to work for both direct to device situations and situations in which devices use proxies. The API from a device directly communicating with the services host is supported, along with a limited set of specific device proxy extensions that deal with transactions and data related to the proxy. All transactions aimed at the devices should look the same.

With reference to FIG. 2, additional portions of an exemplary embodiment are described. The hosted portion of the services, the services host(s) 310, can be located off the user's site 100, 200, and can be located at the manufacturer's or other service providers' facilities 300. A device services edge host 410 handles the transaction and service management for the device services deployed to the field. This includes the management of message queues and provisioning of software modules and configuration parameters in support of the distributed services 140. The edge host 410 is also responsible for the host end of the security and service models employed by the device services system 1.

The edge host 410 also provides connections to service sponsor systems 310. This connects the external device services world to the internal (or third party) services world. The connections to each of the organizations sponsoring the services are maintained through the edge server and can be compliant with security rules and regulations of the entity maintaining the server and host. The edge host insulates the

device 110 or device proxies 220 from having to have direct knowledge of the details of interaction with the back-office complexities of the services involved on the services host 310. This insulation is advantageous in deploying device services in a manageable and robust fashion. Preferably, embodiments present a common services API to the back-office sponsor organizations in order to standardize the basic operations. Specific services can extend and customize the content of the transactions for a given application.

With continued reference to FIG. 2, the devices 110 and user applications 220, such as CWW, can be located in the user's environment 100, 200. This can be a managed services environment as well as a regular user environment. Services and communications are distributed and span from the user's internal systems and network 100, 200 across the Internet or other suitable long-distance connection mechanism 400. Well known web services, as well as future web services, comprise preferred communications mechanisms 130, 230, 330 that are preferably used between the devices/application(s) 110, 220 and the edge host 410, as well as between the edge host 410 and internal services providers 310. The system 1 is preferably built to meet basic IT industry and other standards for its ability to work with existing firewalls both on the user's side (firewall 160) and on the service provider side (firewall 360). No special configuration of the user's firewall 160 is required in order to make this system work properly.

The service supply system 300 is preferably part of the basic supplier infrastructure to provide a robust, well-managed, 24x7 level of service and disaster recovery for all user systems to employ. As indicated above, an edge host 410 can handle the transaction and service management for the device services deployed to the field. This includes the management of message queues and provisioning of software modules and configuration parameters in support of the distributed services. It is also responsible for the host end of the security and service models employed by the device services system. In embodiments, the edge host also connects the external device services world to the internal (or third party) services world. The connections to each of the organizations

sponsoring the services are maintained through an edge server and are preferably complaint with supplier security rules and regulations.

The edge host 410 preferably, in embodiments, isolates the devices or device proxies from having to have direct knowledge of the details of interaction with the back-office complexities of the services involved. This isolation can assist in deploying device services in a manageable and robust fashion. A common services API is presented to the back-office sponsor organizations in order to standardize the basic operations. Specific services can extend and customize the content of the transactions for a given application.

### **Multiple delivery paths**

At the highest-level, the system 1 is designed to enable services 140 to operate directly between the devices 110 and the back-office (supplier) 300 in some systems, or to be operating with the help of a device proxy 220 in others. This ensures the broadest possible deployment as rapidly as possible because the device proxy 220 can quickly bring many legacy devices into the service offering very quickly while user slowly get new devices which are directly enabled themselves. Having both modes of operation is also important because some users will prefer to have a proxy act as a consolidator/clearing house for messages leaving their site rather than from each device independently. In other sites, users may not want to install a device proxy and therefore need the services to be enabled directly. In addition to having each path enabled, it is beneficial to have them work together, because in the end it is possible for users to have both scenarios in place at the same time.

Just as multiple paths can enhance deployment flexibility, it is beneficial to make those paths invisible from the standpoint of the services provider. Preferably, embodiments decouple the devices 110 and proxies 220 from the back office systems 310 as much as possible. A strong abstraction and decoupling of these two halves makes it possible to deploy capability in devices 110 or the back-office 300 in a staged and independent fashion. In addition, if changes need to be made to systems on either end, the

changes will not ripple throughout the overall system if proper abstractions are enabled, enhancing maintainability.

Referring again to FIG. 10, the abstractions throughout embodiments include, at the device level, an abstract device model 122 embedded in the DMA 120. Preferably, the abstract device model 122 is built using the DMTF's CIM as a base. The device model 122 and services platform 124 both reside in the DMA 120, in embodiments. Common services point into the supplier's domain 300 regardless of the services offered. And at the back office/supplier level 300, embodiments use a common API for service sponsors to build and administer services from the supplier back office 300. The common API deals with devices 110 all the same way, regardless of type or connection mechanism.

This architecture in embodiments provides flexible deployment options, such as deployment flexibility in terms of direct device communication to suppliers or communication via a proxy. In addition to that flexibility, the services themselves can be defined so that many of their parameters can be customized. This service customization can include, for example, the data that is sent as part of a remote monitoring service, the time of day or frequency at which meter reads are sent to the supplier. The exact configuration parameters can be specific to the service being offered.

The platform of embodiments is designed to enable the configuration of services to be easily managed. The system allows for the configuration of the services to be specified at the individual device serial number, for all devices at a user's site, or for all machines owned by a user no matter where they are. This management, in embodiments, is done in the back office controlled by the service provider.

An additional part of the flexible deployment options is the use, according to embodiments, of a subscription basis for the services available for devices, as illustrated schematically in FIG. 3. The subscription process can be controlled and managed, for example, by the individual service provider, and the services offered to any given device can be controlled by a combination of user desire and service provider authorization.

Thus, not all devices, even of the same product family, need offer or have installed the same services at any point in time.

There are some activities relevant to DCS. For example, Axeda, Embrace Networks, Questra, and Imaging Portals have been active on the services front. An example of their technological implementations is Embrace Networks' patent application, PreGrant Publication No. 2002-0133581 A1, which is incorporated by reference. However, the prior art lacks provisioning aspects, and there does not appear to be any consequential support for provisioning.

While several companies, such as 4<sup>th</sup> pass, sell general purpose provisioning software, none of the prior art appears to encompass the aspects of the instant invention. For reference, Sun has a general listing of such provisioning software at <http://java.sun.com/j2ee/provisioning/industry.html>. Further, all appear to be pursuing the cellular industry as their target market.

As mentioned above, global telecommunications companies are starting to deliver services over cell phones. To accomplish this, all use a Java standard called CLDC. This released standard describes how Java programs can be run on a small device such as a cellular phone and more importantly how modular programs called Midlets can be added at runtime to a CLDC Java environment.

Although the standard defines the unit of provisioning and how it is to be accepted and integrated on the device side, it says nothing about the server aspects. Because of this, telecoms have either created their own provisioning server solution or purchased one from the provisioning vendors listed above. There is no way to inspect them for alternate solutions because of the competitive environment in this area.

A second relevant standard is called OSGi. OSGi is a Java based, released standard which allows a collection of local, network connected devices to communicate with remote servers and download and run modular services. Compared to CLDC/Midlets, this standard has received much less support in industry.

OSGi also sidesteps the server aspects of provisioning.

A third standard is SyncML Device Management. SyncML is a released standard focused on the details of keeping mobile devices in synch with some server based sources. The focus in this standard is on things like calendars and appointments. In the last year, this synchronization protocol was extended with the Device Management effort to explicitly support the ability to change service settings on a mobile device and to be able to download services to it. SyncML sidesteps the server side of provisioning.

A last standard is unnamed but is commonly referred to as JSR-124. In short, Java programmers use the Java Community Process (JCP) to create and standardize Java Specification Requests (JSRs) as additions and extensions to the Java language. JSR-124 is the J2EE Client Provisioning Specification. J2EE is a standard for using Java in high end, transaction processing. A large and growing market has been growing up around it. Effectively, JSR-124 tries to define a framework within which to express provisioning systems in. Almost all the provisioning startups and many of the telecom companies are members of the JSP. It tries to be common enough so that all provisioning systems can interact with a J2EE system in a standard way but loose enough so that vendors can create alternate, competitive solutions. The standard is in the public draft review stage.

The definition and implementation of a common provisioning model based on a shared user service lifecycle is included in embodiments. A Provisioning Server (PS) 310, the DCS devices 110 that talk to it, and the supplier personnel that interact with it all preferably act according to a shared model for how the provisioning process works. A lifecycle model can be created that defines the roles and responsibilities for each actor that interacts with the PS 310. Based on roles and responsibilities, grammars and commands have been created to allow the actors to accomplish their role based goals.

The architecture and implementation of a provisioning server 900 running, for example, in the services host 310 that meets all the requirements in this section is schematically illustrated, for example, in Table 1 and FIG. 20. Working from left to right in FIG. 20, the first major module is the Service Consumer Interface 901. It is preferably

responsible for all interactions with External Users and External Devices 110, 220. It also preferably isolates the other PS modules from the different protocols that Devices and Customers may use. The preferred protocol in embodiments is Web Services, but in the future may be extended to http, email, cellular or other transmission formats. For incoming transactions, it routes the transactions to the correct internal resource to process the request. For outgoing transactions, it takes the outputs of other PS modules that have been queued for a Device or User and translates them into the required protocol required to interact with the Device or Customer.

<b>Actor</b>	<b>Definitions</b>	<b>Primary Roles &amp; Responsibilities</b>
Service Sponsors	Those organizations within a supplier or other parties that support creation and deployment of services on PS 900	<ul style="list-style-type: none"> <li>• Use Service Developers to develop and deploy required code for the PS and DCS device portion of the service</li> <li>• Create tie-ins between local IM systems and the PS 900 such that commands issued by the service sponsors are synchronized to local state of IM systems</li> <li>• Direct PS 900 to enable and disable service for particular machine in synchronization with local IM</li> </ul>
Service Developers	Those that develop code implementing a DCS based service	<ul style="list-style-type: none"> <li>• Develop code using DCS guidelines</li> <li>• Deploy code bundles making up service to the PS 900</li> <li>• Define service on PS by identifying relevant platforms and other parameter information about the service</li> </ul>
Policy and Preference Setters	Those that define 'rules' that control all aspects of service deployment	<ul style="list-style-type: none"> <li>• Develop the 'rules' that define <ul style="list-style-type: none"> <li>• how default parameters for a service should be configured</li> <li>• how and whether parameters should be uniform for a site, customer, geographic unit, or other grouping derived from service parameter information and/or IM systems</li> </ul> </li> </ul>
External Devices	DCS compatible machines located at user sites accessible directly or through local or remote proxy servers providing a DCS interface	<ul style="list-style-type: none"> <li>• Use the PS to be notified of service life cycle changes (add, delete, modify, upgrade, etc.)</li> <li>• Use the PS to send requests for restores based on local catastrophic failure</li> <li>• Inform PS of relevant machine configuration changes</li> <li>• Inform PS of relevant events or state changes</li> </ul>
Internal Users	Supplier organizations that participate in execution of a service	<ul style="list-style-type: none"> <li>• Send service related transactions to the PS and potentially to other machines or other users</li> <li>• Receive transactions from machines in likewise fashion</li> </ul>
External Users	Non-supplier users that participate in execution of a service	<ul style="list-style-type: none"> <li>• Send service related transactions to the PS and potentially to other machines or other users</li> <li>• Receive transactions from machines in likewise fashion</li> </ul>

**Table 1**

The Entity Management module 902 is a generic PS resource that preferably localizes and isolates entity information from the rest of the server 900. The module holds information on entities such as machines, users, their preferences, and associated location information. For entity information that is not local, the Entity Management

module 902 is the single point of contact with these other IM systems. The module 902 provides a seamless interface for local and network based information.

The order processing module (OPM) 903 is responsible for directing the processing of orders from Service Sponsors as well as those created by the policy & preferences module (PPM) 904. The OPM 903 interacts with required PS modules to achieve the order requirements. The OPM 903 also preferably tracks the status of an order to be able to respond to inquiries from Sponsors.

The registration, authentication, & authorization module (RAAM) 905 is responsible for maintaining the security of the system at all times. The RAAM 905 preferably authorizes all users of the PS and authorizes their ability to execute specific transactions. It is responsible for correctly registering all users, both internal and external. The RAAM 905 does this by working with the entity module 902 to obtain required information. The RAAM 905 is also preferably responsible to work with the service consumer and order processing modules 901, 903 to isolate security related artifacts of transactions.

The service definitions module 906 is responsible for maintaining all definitional information on all services 140 provisioned by the PS 900. Version information, file composition, service inter relationships, product line support, are examples of the contained information.

The service developer interface module 907 is responsible for supporting service developers in their work to develop, distribute, and update services. The service participant interface module 908 is responsible for interfacing with all users and directing service lifecycle and service transaction information to the right resources.

Embodiments apply soft computing techniques, such as, for example, rules and constraints, as a general solution to flexibly model, develop, and examine service policy. The provisioning decision itself is less important overall. That is, given a device 110 that needs a service 140, the PS 900 determines whether it is allowed, whether there is a bundle (the collection of code files that make up the service to be installed) that is

compatible with the device 110 operating parameter information (model type, OS version, etc.), which of a plurality of bundles should be selected if there are a plurality, and what the parameter settings (if any) for the service 140 should be. Generally, in embodiments, code can not be written that implements “business rules” that can be used to resolve the questions above. Coding would be required for every change of a rule, the rules would not be directly inspectable by policy makers, and it would assume that each question is separable from the others. Further, it assumes that there is a single policy maker that determines the answers for all the above questions. Thus, an alternate solution must be, and is, provided, in embodiments.

The introduction of an appropriate constraint or rule system provides advantageous benefits. Coding is dramatically reduced as the “rule” is entered at a higher level of abstraction. Additionally, the rules are inspectable by policy makers who may not be comfortable with computers or programming. Further, knowledge implemented as constraints and rules relating to each question can be more easily combined and separability doesn’t need to be worried about. Constraints and rules use supports the reality of multiple policy makers that participate in the decisions of the above questions. Interfering rules and constraints based on differing groups participating in a value chain can more easily be identified and resolved.

The ability of the provisioning server 900 to use policy based knowledge to identify the correct bundle and parameters is advantageous in several situations. For example, this ability is preferably applied when the PS 900 has received an Add Service Request and needs to compute the answers to the questions above. Additionally, the PS 900 employs this ability when a Policy Maker for any service has made an update to the policy knowledge. The PS 900 can compute the impact of that change, addition, or deletion to the existing relevant devices actively connected to the PS 900. Then the PS 900 can generate the necessary change requests to the impacted devices 110 to achieve the goals of change and use the change in all future Add Service transactions. When the PS 900 is notified of a Configuration Change from a device 110, the PS 900 decides

whether the device's services 140 and/or parameters should be changed because of that change. If necessary, the PS 900 can generate change requests for the device 110 as required by the policy knowledge.

Policy Setters can define uniform service versions or parameter settings based on Internal or External Customer requirements through use of rules. This uniformity may be defined at a user level, a site level, a machine category, or any other relevant grouping

To summarize, the service subscription and deployment method includes identification by a user or user DMA 120 of a service offering 140 of interest and a request for activation of such service (block 501). During a scheduled check in with the edge host, or during a special connection for the purpose, the DMA 120 sends a message for the supplier system 300 regarding the interest and requested activation. The supplier system 300 retrieves the message from the edge host 410 and applies business rule and work processes to determine user eligibility (block 502). If the user is approved, the supplier system 300 notify the edge host 410 that the requested service 140 can be added (block 503). The next time the DMA 120 checks in with the edge host 410, it receives the message that the service 140 can be added (block 504). The DMA 120 then activates the service 140, downloading and/or installing it if necessary (block 505). The new service is then deployed and running (block 506).

Sales of services can be done through a plurality of channels. This process is preferably owned by the sponsoring organization (the supplier of the service) and is done in whatever manner the sponsoring organization chooses. It can, for example, be done from the device if desired.

Once the sponsoring organization is notified that a particular user would like a service to be enabled on a given device, embodiments provide that the sponsoring organization applies whatever business rules and billing/invoicing processes it requires to comply with an applicable business model for that particular service. If the sponsoring organization determines that the device can be permitted to provide the specified service to the user, the sponsoring organization uses a common services order/entry API on the

edge server to officially place the order. This can, in embodiments, generate a message that can set deployment and configuration of the desired service into motion.

Messages are preferably queued for delivery, and the process waits until delivery of the messages occurs. Once the requesting device or device proxy gets the order message, the systems are configured, additional software is downloaded if required, and the new services are started. The service sponsor preferably has the ability, via the system according to embodiments, to turn services on and off as needed based on whatever criteria the service sponsor determines is necessary. Services are preferably written to be device independent. The Common Information Model provided by the Device Model Agent provides a device independent representation of the common data and methods in embodiments. Services are configurable since all users do not have the same requirements. Having configurable services accommodates variation in requirements and operation that may be required. Services are dynamically loadable to enable rapid deployment of new services to users with devices already deployed in the field. And services have a lifecycle to enable management after they are initially deployed. Examples of lifecycle transactions include, but are not limited to, add service, delete service, modify service, sync services, device registration, and proxy registration.

The DMA 120 is defined, in embodiments, to enable the embedded computational power, data, and functions of the devices 110 to be accessed and used in services 140 that are deployed in a common fashion. An embedded agent 122 and service platform 124 enables embodiments to support local operation of services 140 that play into the overall system 1. This provides the common connectivity, service manager, common data access and methods, and secure communication to the service provider/supplier in support of services offerings.

Given the system, components, methods, and embodiments described above, there are a number of ways that the system can be deployed. This deployment flexibility is a significant advantage of this system and has implications on the detailed designs of the components and behavior models that the system follows. All of these deployment

options can be instantiated simultaneously by embodiments given the abstractions and modularity defined. It is possible that in many user installations more than one option can be deployed to ensure complete coverage. FIGS. 4-9 and 11 show several exemplary embodiments representing possible deployment options for systems according to the invention.

Deployment A, an exemplary embodiment seen in FIG. 4, is a preferred embodiment for smart devices as are currently shipped by some companies, such as Xerox® Corporation. It can limit the amount of infrastructure required of users to support deployment of services 140 and provides the simplest implementation. It need not require additional hardware or software to be installed in the user's environment, though devices 110 must be fitted with the functionality of the DMA 120, including the services platform 124, if they are not already part of the devices 110. This embodiment is not likely to address many machines already in the field unless the device software is upgraded or another method is employed to give the fielded devices the DMA and services platform. While communications between the device, via the DMA, and the back-office host are substantially independent of the physical medium, preferred embodiments employ the user's network and the user's Internet access to connect back to the supplier host system. Other communications schemes, such as, for example, local wireless, long distance wireless, telephone, wireless telephone, and satellite telephone can of course be used as well.

As seen in FIG. 4, each device 110 includes its respective DMA 120 and runs its own services 140 in its own services layer 141 facilitated by the DMA 120. Management and other applications 220 can be employed on another machine 200 that can also be in the user's environment 100 or can be elsewhere. The devices 110 preferably use web services 250, such as HTTP, HTTPS, and SOAP, to communicate with the supplier 300 and a services host 310 therein. The services host 310 includes services 320 and host systems 340 that can assess communications from the DMAs 120 and deploy services 140 when appropriate.

Deployment B, another exemplary embodiment seen in FIG. 5, enables already fielded devices and devices produced by third parties who do not have the required technologies embedded in them to support device services. While multiple devices can be handled in such a manner, this description will focus on one such device for simplicity. In this case, a relatively small add-on component 115 is added to the device 110. The add-on component 115 contains necessary software and the DMA 120, as well as one or more connections to the device 110 to enable the add-on component 115 to gain access to the internal data and functions of the device 110. With the add-on device 115 attached, the device/add-on component combination looks like a completely enabled device, as in deployment A seen in FIG. 4, to the rest of the services infrastructure and back-office systems. This provides device services according to embodiments for legacy and third party production equipment. The add-on component 115, with the DMA 120 and its attendant services environment 124, then communicates with the supplier 300 via web services 250 as in deployment A of FIG. 4.

Deployment C, a third exemplary embodiment seen in FIG. 6, uses a proxy configuration in which an application 220 capable of acting as a proxy runs the services for at least some of the devices 110. Devices 110 that do not themselves have the required software enablers embedded, such as the DMA 120 and services platform 124. However, an application 220 acting as a services proxy for the devices can communicate with the devices 110, such as, for example, via LAN, phone, wireless, or other communications media. The basic proxy implements the services APIs 140 for a selected set of services 140, but preferably does not use the full DMA 120 and standard dynamic services deployment method to the devices 110 themselves since these features can not be supported with the legacy devices. This deployment is also limited by the richness of the connection between the simple proxy and the device: if data or a function can not be accessed remotely, then services that require them can not be deployed.

Deployment D, a fourth exemplary embodiment shown in FIG. 7, is a more advantageous form of proxy configuration. This embodiment enables devices without the

required embedded software enablers (i.e. the DMA 120), but that can communicate in other ways, such as, for example, via LAN, phone, or wireless, to participate in the services deployment system. The devices 110 communicate with one or more applications 220 that act as a services proxy for the devices 110. The services proxy is a DMA enabled proxy that can host a DMA 120 for each device 110 communicating with the services proxy. Additionally, the services proxy can manage the DMAs 120 for the devices 110 with which it communicates. This enables the services 140 to run in substantially exactly the same way on the services proxy as they would if the services 140 were running directly on the devices 110 themselves. This also enables additional local applications to be written on the services proxy that can take advantage of the DMA 120 and the common information model representations of the data and functionality of each of the systems. This can greatly simplify applications since they can be hidden from the implementation specific to each device and only have to build to the common representation of data and methods in the CIM. This is the same advantage that the services gain when written against the CIM and DMA. Additionally, portions of the DMA that can manage multiple instances of the CIM and services can be instantiated once and used to manage the DMA proxy of multiple devices. That is, the full DMA need not be replicated for every proxied device, which can make this embodiment more efficient than just dropping all the DMAs for the connected devices onto one server.

Another aspect of the services proxy embodiments is that portions of the DMA that can manage multiple instances of the CIM and services can be instantiated once and used to manage the DMA proxy of multiple devices. Thus, the full DMA need not be replicated for every proxied device; rather, one DMA can be used for plural devices. This makes deployment more efficient than simply dropping one DMA for each device onto one server.

In a particular version of deployments C & D, embodiments cover the deployment of a device proxy for a printer directly connected to a personal computer. The proxy can be hosted on a user's computer, and a printer, such as a printer connected

via a parallel interface, is the device with which the proxy interacts. In embodiments, the proxy can also connect to the print driver for the directly connected printer as an additional source of data to populate the DMA or services interface. The computer can host the DMA and, to the extent supported by the direct connection to the device and the local instrumentation via print driver or other access mechanism, the directly connected printer looks networked from a services and systems management point of view.

Deployment E, a fifth exemplary embodiment seen in FIG. 8, comprises a local variant of the exemplary embodiments seen in FIGS. 6 and 7. Services can be offered locally, that is, within a substantially self-contained site, in a fashion similar to Internet-spanning embodiments. Such an embodiment uses the abstraction of the DMA 120 to enable more consistent management and service offering implementations to local devices 110. While this lacks the connection to back-office service providers 300, the services 140 can be unique for a user or simply self-contained for security reasons. Management of local services 140 and devices 110 can then be moved from a centralized locale for all devices 110 to a more localized domain. Users can, for example, assume the role of supplier, if so desired, in such an embodiment by running the equivalent of a back-office on their intranets, including application servers, and, depending on user security requirements, edge hosts. This will increase the complexity of the maintenance and support of such a system if offered by a third party, but is a possibly useful configuration given the abstractions defined.

A further exemplary embodiment, Deployment F in FIG. 9, enables multiple application servers 310 and/or multiple edge hosts 410 receiving communication from enabled devices 110. Deployment F is an embodiment that can combine, for example, elements of deployments A, B, D, and E. The services 140 can be written such that they describe everything required for the services 140 to check in in an appropriate fashion with appropriate application servers 310 via the appropriate edge hosts 410. In addition, the services host 310 to which the edge host 410 connects the devices 110 is not limited to any particular services host or supplier 300, but can connect to any suitable parties to

offer services, as long as the services API presented by the edge host 410 allows such connection.

### **The Device Model Agent**

The Device Model Agent (DMA) 120, as discussed above and as seen, for example, in the schematic illustration of FIG. 10, is an enabling component of the end-to-end system 1 according to embodiments. The DMA 120 can be embedded in devices 110, add-on modules 115, and/or device/services proxies to provide a common device model 122, a CIM API 123, and a device services environment 124 in which services 140 can run. The DMA's role is to provide devices 110 with the capability to actively participate in business process and services that surround the devices throughout their lives. It combines aspects of the Common Information Model Object Manager (CIMOM), from the Distributed Management Task Force (DMTF), and a novel environment for the operation and management of embedded and dynamic services. The agent is responsible for local operation of services and the management of information represented in the CIM. The agent interacts with the device, services (both local and distributed across a networked environment), and other distributed system components.

The DMA provides the device independent CIM API as specified by the DMTF, but also provides a device independent Service API. As a software agent, the DMA can engage in autonomous and adaptive behaviors, either initiated locally or through interaction with other distributed components. The DMA can also, for example, react to events in the device and the environment, again either locally or distributed, and, in embodiments, can engage in self-management of services and actions. In a preferred exemplary embodiment, the device independence of the DMA is extended through the use of, for example, JAVA and the J2ME small footprint JAVA standards. Of course, the DMA is not limited to this particular implementation and could be assembled in any suitable software structures with varying degrees of complexity and difficulty to provide all features. This exemplary embodiment of the DMA advantageously uses the J2ME

Connected Device Configuration with the Foundation Profile to enable the broadest range of devices from large system components with many resources to small systems with limited resources. Again, the Device Model Agent is not limited to this implementation, and many others are possible in variants of JAVA or other programming languages as required by the device in which it resides. The J2ME environment can ensure that the DMA software is device independent and reusable across device and product platforms. J2ME also offers support for networked and distributed systems, embedded security capabilities, and support for dynamic download and operation of code.

Preferably, embodiments include extension of the device independence of the agent through the use of a platform-independent standard, such as, for example, the JAVA and the J2ME small footprint JAVA standards. Of course, the agent is not limited to such implementations and could be assembled in any software structures with varying degree of complexity and difficulty to get all the features. Embodiments of the agent using a J2ME Connected Device Configuration with the Foundation Profile can enable a broad range of devices, from large system components with many resources down to small, embedded systems with limited resources. Many other embodiments are possible using variants of JAVA and other programming languages as required by the device in which the agent is to be embedded or which the agent is to represent. The J2ME environment is a preferred environment due to its ability to ensure the agent software remains substantially device independent and substantially reusable across device and product platforms. In addition, J2ME includes support for networked and distributed systems, embedded security capabilities, and support for dynamic download and operation of code.

In addition to the benefits described above, the DMA provides the ability to hide multiple, disparate sources of data behind a common provider API. This further abstracts the details of the device from the software agent. In embodiments, four separate sources of data can be unified behind the common provider and CIM so that the services need not know the details the data sources. For example, EPSV, PWS, CAN Bus, and

Web UI, data can be managed in this fashion. A set of tools can also be provided in embodiments that enable the provider layer and the CIM contained in the Device Model Agent to be easily customized for a given product or device. This encourages reuse and speeds release because programs adopting or maintaining the Device Model Agent need only be concerned with the mapping of CIM element to the source of data and not the management of the entire Device Model Agent.

An enabling feature of the end-to-end architecture of the system components is the inclusion of, in embodiments, an appropriate abstraction of the communication methods employed between various distributed components. This abstraction is preferably applied to the physical connection mechanism, as well as to the protocol level up through session level. Such abstraction at both levels helps to hide the details of the communication method from the distributed components, allowing them to focus on the operation of the services and decoupling them from changes in the communication media or protocols. For example, this allows the system to use email over a wireless link or Web Services over a dedicated Ethernet link without the services themselves caring which is used.

This type of abstraction is new to devices and provides several important benefits. It provides for flexibility in deployment of the system components for any given user. Information on the Quality of Service that can be expected from any given combination of physical and protocol, up through session layer, connections can be carried. The system can have a component on the host/back-office side that monitors the Quality of Service for various configurations to assess the effectiveness of a communication link for providing the quality of service required by a given service offered to a particular user. This is an element of the provisioning and self-monitoring portions of the overall end-to-end system.

The communication abstraction also provides some fault tolerance. If one connection mechanism goes down for some reason, the communication module can detect that and replace the failed connection with another working one without the rest of

the system knowing other than the fact that potentially a change in the quality of service has occurred.

In embodiments, services can alternatively be “hardcoded” into a device or proxy. This means that many of the management functions associated with dynamic adding and deleting of services is not required. The embedded portion of the service that is to run at the device must be compliant with the web services transactions between distributed components. This enables the back-office to effectively treat “hardcoded” services the same way as full, dynamic services in the system.

Hardcoded services can be enabled by back-office subscription. This enables the service provider to control the particular services enabled on any given device, which gives the service provider the flexibility to determine how the services offered will go to market based on business need. For example, the services can be part of a package, offered for free, be offered for a price, require renewal, be offered on a trial basis requiring another transaction for full service provision, etc.

Hardcoded services preferably share a common underlying set of behaviors and specific extensions for their particular requirements. Preferably, the services have components that work together, but run on the devices themselves in the embedded services platform, on the intelligent proxy, and/or in the back-office server. Though hardcoded, these services can be configured and managed by the service lifecycle management system in the supplier/service provider back-office.

The types of standards embodiments preferably use include the Distributed Management Task Force (DMTF) Web Based Enterprise Management (WBEM) and Common Information Model (CIM). As described above, the CIM provides embodiments with device model and abstraction to enable services reuse. Additionally, embodiments employ web services, XML, various versions of HTTP, and SSL. Embodiments can also use server side certificates from, for example, VeriSign, which enables communication across firewalls and the Internet. To enable application environments in the device and in the back-office, embodiments can employ, for example, Java 2 Micro Edition (J2ME),

Embedded Virtual Machine from Insignia Corporation, Java 2 Enterprise Edition (J2EE), the BEA WebLogic 7.0 Application Server Technology Suite, and Oracle8i. Of course, these are only examples, and additional components can be used where appropriate. Further, it is likely that new components will be developed that are not currently foreseen and that can be added to the system of embodiments, which components fall within the scope of embodiments. The services provided, their lifecycles, and the extension of the DMTF CIM for specific products are examples of new technologies within embodiments.

Embodiments further enable the rapid addition and roll out of new services to already deployed systems. For example, say that soon after the launch of a new product a new diagnostic service is developed based on lessons learned form the first three months of its operation in the field. The exact nature and behavior of this service could not have been anticipated when the product was launched, so the diagnostic service would not have been included in the launched product. Embodiments allow such a diagnostic service to be added to installed devices at substantially any time.

Embodiments contemplate the service model and internal specification of what a new “service bundle” should include. Thus, in addition to the permissions and configuration information for a service, new code can be downloaded if needed to add a new capability to an existing device in the field. This feature can be used in conjunction with, according to embodiments, an embedded services platform on the devices that are designed to accept the new functionality easily. In addition, when used in conjunction with the embedded services platform of embodiments, the new code for the new services can be reused across platforms because of the device independent abstraction provided by the embedded CIM in the DMA. For devices without such a platform, new code can still be added as, for example, a more specialized software download service for patches and upgrades in the field, but the code to enable those services will most likely be platform specific and therefore less reusable.

This system of embodiments can offer diagnostics routines and other services in a way that is very flexible for the device platform. To the service provider in the back-

office, such enabled devices look like every other DMA enabled device according to embodiments. In addition, all the services for the device family that run locally on the device internal platform can still communicate directly back to the supplier systems rather than through an intelligent proxy.

Another variant in deployment is to fully embed the DMA into the product itself. This implementation is in a way very similar to the Example 1 implementation in that they are both DMA enabled platforms. For this example however, the small footprint DMA services platform is embedded into the product and communicates with both a Print Station Interface Platform (PSIP) and with an embedded device controller. The limited resources required by the small footprint system is acceptable to that product and development and integration of the required interface components is relatively easy.

The reusable DMA is a “drop-in” to systems that already have a JVM. The small footprint DMA is not a drain on the system resources and can greatly speed the enablement of such platforms.

### **Automated Meter Reads**

Another example of using the deployment flexibility built into the embodiments is seen by looking at the system from the perspective of an end-to-end service. In this case, the service is automated meter reads. This service focuses on acquiring the monthly or quarterly meter reads typically received via phone calls, faxes, emails, or web entry without a human in the loop. This can increase both the accuracy and timeliness of the reads, save time for users, and enable suppliers to improve invoicing and billing.

Since the data required from the devices is small and is already largely available, an intelligent proxy can be employed, which can facilitate the participation of all SNMP enabled products. This, used in conjunction with devices that are DMA enabled but not fully SNMP compliant, means broad coverage can be achieved rapidly. Again, the abstractions and the system modularity in this case are significant. The back-

office system doesn't need to know which way the devices have contacted the supplier (directly or through a proxy); all it needs is the device's serial number and it can then request meter reads when they are due. This decoupling of the way devices are enabled to participate in the services and the requests made by the back-office service provider is an advantage in providing deployment flexibility.

### **Early Warning System**

In embodiments, a reporting system, a remote monitoring service, and other remote services are combined to assemble a set of tools to support more testing in the field. The underlying systems and data collection services can be complementary to data collection systems that rely on human observation and reporting. Together, the combination of systems provides a much greater, integrated set of knowledge upon which engineering teams can base product problem resolution activities. In addition, the common model for data collected from devices in the field creates a mechanism for deploying reporting tools and basic performance reporting that can be used across platforms.

### **Premium Remote Assistance via Remote Control and Device Services**

One of the basic principles in embodiments is that the devices themselves should take an active role in their own lifecycles and support. This works in a number of trouble or status reporting situations. It may even work with an embedded diagnostic agent in the device that can monitor system performance and make software or configuration changes automatically in order to keep the system running well in the field. However, many problems that users experience are related to user problems and operational errors as much as they are related to device failures. In addition, as we all know, since marking machines are complex electro-mechanical systems, they cannot always be repaired remotely.

To address the operational support needs of devices in the field and to support new ways of working with the operators on site, a remote UI and a human to human support system are combined in embodiments. Support automation solutions can be complementary parts of a premium service and support offering. It automates data collection and remote monitoring as well as offers many remote services described above. The combination also provides an excellent way to work directly with the device operators via a shared UI to help them when additional training, problem resolution, and software tweaks are required.

### **Connectivity Trade-offs:**

Some exemplary options for the communication link between the devices in user sites and the back-office are shown in Figure 41. There are three primary options labeled A, B and C. Notice that only options A and C complete the connectivity between the devices and the supplier back-office on their own. Option B needs to connect to A or C to complete the link back to the supplier.

A summary of the pluses and minuses of each option are in Table 2.

	<b>Wired Connectivity</b>	<b>Wireless Connectivity</b>	
	<b>Option A:</b>	<b>Option B: Local</b>	<b>Option C: Long Distance</b>
+	<ul style="list-style-type: none"> <li>• Available to all network devices in sites with Internet access.</li> <li>• No additional user or supplier cost for use of LAN and Internet</li> <li>• Phone line can be added regardless of printer / stand-alone option</li> </ul>	<ul style="list-style-type: none"> <li>• Use wireless connectivity will eventually be available in printers</li> </ul>	<ul style="list-style-type: none"> <li>• Technology exists for both cellular and two-way pager options.</li> <li>• Can be purchased off the shelf from several companies or optimized specifically for supplier from standard technologies available.</li> <li>• Potentially good answer for unconnected devices if leverage supplier's existing pager / cellular service plans.</li> <li>• User feedback has been positive for this option for unconnected devices without any costs being passed on to them.</li> </ul>

<ul style="list-style-type: none"> <li>• Some additional traffic on users network</li> <li>• Unconnected systems not covered by LAN option</li> <li>• Phone line option is added expense to enable and operate</li> <li>• Phone line connectivity has proven to be difficult to maintain.</li> </ul>	<ul style="list-style-type: none"> <li>• Wireless capability not yet available in non-connected printers yet.</li> <li>• Wireless access points for connectivity to rest of network may not be present</li> <li>• Local wireless still requires either A or C to be present to complete the link to supplier.</li> <li>• Need for wireless may be temporary as more devices become connected.</li> </ul>	<ul style="list-style-type: none"> <li>• Two-way pager and cellular add-on components costly.</li> <li>• Users have security concerns about networked devices that are also connected wirelessly to another network.</li> <li>• Bandwidth limited vs. wired connectivity</li> <li>• Need for wireless may be temporary as more devices become connected.</li> <li>• Add-on boxes are likely to require unique communication interfaces to connect the box to each type of printer.</li> </ul>
--	--	---

**Table 2**

All connectivity options preferably reuse the same back-office infrastructure even though they may enter the supplier via different mechanisms.

All options are attractive because as a group they can provide additional flexibility for deployments that will meet a variety of user requirements. The preferred method of connecting, when feasible, is Option A - Wired connectivity via LAN and Internet. This is the option of least development investment and least operation expense. In the short-term this is especially important while the value of the services are being proven and resources need to be focused on initial services development and delivery – not additional ways to connect to devices. It does not, however, address unconnected devices that will initially be left out of the services if only this option is pursued. For the time being each service will need to consider how to manually include non-connected devices in the offerings.

The next preferred method of connecting is Option C - Long-distance wireless via cellular or 2-way pager technology. The system can work in this configuration seamlessly with wired devices, and having capability available would enable some user problems to be solved when they come up. However, there are some challenges with deploying the wireless capability on a large scale over a large number of products. For example: developing a number of different add-on modules to be compatible with the

very wide array of products in the field could be costly since few systems have the same interfaces to access detailed device data and operations. Additionally, the added expense of adding wireless connectivity and communications costs may be prohibitive until several services are available to use the connection. Simple, more easily deployable wireless configurations have inherent limitations on the number and types of services that can be offered, making it harder to justify the cost. Finally, users voiced concerns with networked systems that also had wireless connections since this is a way that suppliers/service providers or another party could bypass their firewalls and potentially access other resources on their network.

Finally, Option B - Local-wireless connectivity. This method can be used depending on how the local wireless connectivity technologies integrate into our user's environments and printers specifically.

#### **Supporting End-to-End Infrastructure for Device Services**

A supporting end-to-end infrastructure for connecting devices in user sites back into legacy systems and business processes is required. The end-to-end system shown in FIGS. 1 and 2 is an initial exemplary embodiment of an end-to-end infrastructure. It supports the basic dual-mode of device participation (direct and via a services proxy), uses the initial service communication and subscription models, and employs a common entry point for services data and actions via an edge server hosted, for example, in the supplier environment. The edge host can be partitioned in a manner suitable for additional embodiments, but can also be physically hosted on one system, minimizing start up costs while penetration and adoption ramps up.

Enabling Technology Evolution	Infrastructure Evolution
<ul style="list-style-type: none"> <li>• High-end devices can act as services proxies for other devices they find within their environment if the users chooses</li> <li>• Device can participate in a combination of services directly or through proxies.</li> <li>• Secure communication initiated by either devices or supplier can be deployed services that require high level of service guarantees.</li> <li>• CWW can communicate with new devices via the CIM protocol vs. only SNMP</li> <li>• Device can participate in services via wireless connectivity directly from them to supplier.</li> </ul>	<ul style="list-style-type: none"> <li>• Edge Host focuses on service provisioning and transaction management</li> <li>• Supplier maintains common DataMart where CIM data from all devices is stored and is accessible by a variety of internal supplier functions.</li> <li>• In addition to sending data to the DataMart, data/events are routed based on subscription and services needs directly to the service sponsors</li> <li>• The physical split between the Edge Host, database, and service specific</li> <li>• Authentication /Authorization of devices and communication is handled centrally via supplier systems.</li> <li>• Provisioning and software/tools for services to define the business rules which describe how devices need to be configured to participate are established and used to deploy new services quickly.</li> </ul>

**Table 2: A sample of the Technology and Infrastructure.**

Each of the areas listed in Table 2 represent areas of technology development or areas where third party COTS systems need to be acquired and explored. They also represent areas where the full requirements for the technologies are not yet known.

As described above, printing products not originally designed to support user assisted self-help programs, device-centric services, and/or remote monitoring for ECAT sometimes find that such offerings are important to speeding initial delivery and to continued success of the products. A need of such products is to receive daily (or at some other period) reports from devices in the field as to their state and how they have been used by the user. We have called this service Remote Monitoring. This is important as it allows the program team to identify problems earlier in the field and provides important information to enable, sales, marketing, and support to improve their outputs as well.

One solution to this is to offer the Device-Centric Services (DCS) Device Model Agent (DMA) 120 on the controller and connected locally to the IOT from there. The add-on component or Customer Services Platform (CS Platform) 115 is the solution to this need. The CS Platform 115 can take the form of an embedded system that connects locally to the IOT through one or more of several existing interfaces, unifies the view of that data and functionality, and provides a local UI for operation, management of functionality locally, and the services platform 124 and APIs for remote connectivity and

device-centric services. The CS Platform 115 is a product embodiment of both the DMA 120 and the embedded services layer 141 enabled by the services platform 124 in the Device-Centric Services framework.

With reference to FIGS. 12, 16-19 and 21, the CS Platform 115 can preferably take the form of a networked, embedded personal computer. Additionally, the add-on component can take the form of a headless box. In whatever particular form, the add-on component 115 is connected to the IOT via at least one physical interface. The UI for the CS Platform 115 is available at any browser on the local network and is served by an embedded web server 130 in the CS Platform 115. In a preferred embodiment the user would use the browser on their DFE as the local UI for the CS Platform 115. The CS Platform 115 is preferably networked and configured, just as any browser is configured, to know the local network proxies, firewall passwords, DNS server IP addresses, and so forth, so that it can connect to the edge server 410 which is available on the Internet 400. When running, the CS Platform 115 will use this connection to check for messages and instructions and will send required data in support of subscribed services 140 as well. The edge server 410 manages the queues, messages, services, and transactions associated with the end-to-end operation of the device services.

Preferably, the CS Platform 115 is a low cost, embedded personal computer based platform with a motherboard 701, and an embedded software operating system 704, such as Linux, though other operating systems could be used. The add-on component 115 can be customized with hardware, such as an auxiliary input/output and static memory board 702, but such customizations are preferably minimal to keep costs down. The component 115 is designed to enable the internal hardware platform to change over time to follow the minimum generic personal computer value curve which can reduce the cost of the platform by 2/3. Memory, such as compact flash memory, for example, can be used as an internal storage medium 703, which has improved reliability over hard disk drives. The use of compact flash memory also lowers the cost of upgrading the CS Platform 115 if new services 140 to be deployed require additional storage resources, yet

the compact flash memory appears to the system as a normal hard disk drive. Further, the use of standard personal computer technology in the add-on component 115 enables rapid revision to follow cost curves and trends and also ensures that standard add-on technology (for example a web camera) is compatible with the platform.

Examples of connection paths between the CS Platform add-on component 115 and the IOT of a device include Electronic Partner (EPSV) 712-714, a Fuji Xerox protocol and interface; RS422 and/or RS232 serial port 715, 716; PWS 717-718, the connection used by the customer service engineers to connect their service laptops to devices; CAN bus connection 719-721; and USB (not shown). Additional interfaces, such as a proprietary interface to the digital front end, could also be monitored to provide additional data for services and system management activities. Other connections fall within the scope of embodiments, as well, especially since the CS Platform 115 is preferably designed in such a way as to not be limited to these connections.

A router 730 is preferably included and responsible for managing the multiple information sources and handling preemption of some activities given another connection becoming active. Communication is thus mediated in such a way as to allow communication to occur without data corruption problems.

An embedded software system preferably provides the flexible components in support of both locally hosted functions, such as the diagnostics routines described here, and services that can be dynamically added and configured. Embodiments thus contemplate a system component based on the Device-Centric Services platform and embedding the DMA 120, and an embedded JVM and web server to enable the CS Platform 115 to act as a local enabler for the system to actively participate in device-centric services.

The embedded DMA 120 enables services 140 to be offered directly from the device 110 regardless of its digital front end and/or ability to run the DMA 120 on its own. This enables the device 110 to participate actively in services offerings via the DCS services model. The add-on component 115 also provides a programmatic interface for

new services 140 to be built around the system, enabling rapid and robust solutions integration with the product. Further, the inclusion of a web server 130 in the add-on component 115 allows web services transactions and services directly between the CS Platform 115 as an interface for the IOT and remote service offerings.

A customized IOT diagnostics offering 740 geared towards a trained user rather than the customer service engineer provides easy to use, globalized UIs for predefined diagnostics already offered by the IOT. The diagnostic routines can, for example, optimize toner density levels and obtain consistent image quality (MaxSetup 741). Other services 742 that can be offered include Belt Edge Learn, a routine that learns the edge of a new intermediate belt to improve lateral registration and belt steering performance. The purpose of Belt Edge Learn is to track the belt movement using the two Belt Edge Sensors. Using data received from these Sensors, the IOT automatically adjusts using the Belt Tracking Roll/Motor/Sensor to ensure that the Belt rotates without any inboard/outboard movement. More services can include RegiCon, a set-up routine that sets up the complete image on image registration system found in the IOT, and Halftone, a set-up routine to adjust the halftone densities printed by the system. Printing a halftone pattern places a user-definable level of constant tone over the whole page. The halftone pattern itself is used to diagnose problems, as almost every image quality defect will show up in a halftone pattern.

The CS Platform add-on 115 preferably employs a web based UI through an embedded web server. This saves hardware cost on the CS Platform 115 itself and instead uses the monitor, keyboard and mouse hardware associated with the digital front end of the device 110, which is almost always present and networked. It is also accessible from any other networked PC with a suitable browser on the local network. Such a UI offers increased ease of use and extendibility for new services and capability over time. The cost of providing a GUI just for this application would be prohibitive. The web based UI can include, in embodiments, context sensitive help and links to a call center and other support sites, making the system much easier to use. The UI can be available at any

connected browser on the user's LAN, including hardwired networked personal computers. Further, wirelessly connected personal computers or handhelds with compatible browsers could also be used as a UI if a wireless access point is connected to the CS Platform 115 directly or installed on the user's network.

The Device-Centric services add-on component 115 of embodiments preferably comes equipped with several services pre-loaded and authorized, though this need not be the case. The CS Platform follows the Device-Centric Services model for subscribed service offerings. The PDT has made the decision to enable a basic set of services. Additionally, the component is preferably enabled for software download and remote upgrade of the CS Platform firmware via notification from remote site.

The system, through the synchronization service, will preferably periodically check in with the remote DCS host 310 or 410 to see if new transactions are waiting for it. One of those could be that new software is available for the system. If so, the user can be notified via an upgrade status screen available from the administration tab. The user also has the option of manually checking for updates via a refresh status button on a software upgrade screen. If an upgrade is available, the user has the option to accept it. If accepted, the software download process automatically downloads the required updates, installs them, saves the older version, and reboots the system.

The component 115 provides secure, encrypted communication back to the supplier in support of eService offerings. Diagnostic routines of embodiments, even though completely local in operation, are treated as services. They can then be controlled via the services subscription model used for all services. This allows the functionality of the CS Platform to be effectively turned-off when it becomes necessary to do so.

An initial set of services can preferably be offered to the user by the system. Such an initial offering can include, for example, Automated Billing, Automated Supplies Replenishment, and Remote Monitoring. Automated Billing is preferably a subscribed service that on demand or automatically reports the required billing meter to the supplier via the Device-centric services infrastructure. Automated Supplies Replenishment, as the

name suggests, is preferably a subscribed service that tracks toner usage, area coverage, and toner bottle change events in order to supply the supplier with the information necessary to ensure the timely and accurate delivery of meter supplies to the user's site without human intervention. Remote monitoring is preferably a service that periodically gathers up a configurable set of data found in the system, models it in a standard fashion, and publishes it back to the supplier. Examples of the type of data found in this service include billing meters, IOT faults, media path jams, image area coverage, media usage (weight, size, and type), feature usage, toner status, simplex/duplex quantities, media tray usage, reduction and enlargement, copy modes, and High-Frequency Service Items status.

An additional set of services can be embedded in the system to ensure proper system operation. For example, DMA Housekeeping Service, Health Monitor, DMA to IOT communication status Monitor, and Services Synchronization Service, a service that periodically checks back with the remote portions of the DCS system to see if there are new instructions or activities the DMA should be doing.

To ensure security, the add-on component 115 employs in embodiments standard, secure web data transmission technologies and certificates. For example, VeriSign certificates, RSA encryption, SSL, and related technologies can be employed. Additionally, the add-on component 115 can provide a detailed transaction log allowing the user to inspect all the messages sent from the device. All transactions sent from the CS platform 115 can be logged in XML form before they are packaged for transmission and encrypted. This provides another layer of inspection capability by the user to increase confidence in the supplier's statements that we sending only what we say we are.

Three levels of authorization can be invoked before data is sent to the edge host 410, in embodiments. It is expected that contractual agreements will state that data will be sent automatically and the user will have the ability to inspect the transmission logs. Options are designed into the system to accommodate multiple levels of authorization for users who require different agreements to be made. The levels can include Audit and Log, in which records of all transactions are kept in the transaction

Log; Simple Notification, in which a user representative is notified via on screen message, email or some other mechanism when a transmission to the back-office is accomplished; and Approval Before Sending, in which a queue of messages to the back-office 300 is maintained and the user representative is notified when the queue is not empty. In Approval Before Sending, the user representative can inspect the messages if desired and can then OK the sending of the data. The default authorization level is configurable, though the preferred shipped default level is Audit and Log. Previous assisted self-help tools and even the diagnostics access on the IOT itself had one password for all functionality. There was no way for the previous system to accommodate multiple people roles and manage passwords accordingly.

In embodiments, the multiple roles enabled can include Technical Key Operator (TKO), Customer Service Engineer (CSE), and System Administrator (SA). A system for configuring the access for any given role is provided via the web based GUI. Passwords are preferably initially set to common passwords individualized for each role. The system of embodiments contemplates allowing the SA to configure his or her own password and manage the passwords of the TK); enabling networked role based password management using standard IT industry processes, protocols, and procedures, and enabling remotely authenticated login and password management for any or all roles. Remote login may be especially attractive for CSEs who want to use their same password on any CS Platform 115 that they visit. Authentication for remote login can be password only, a combination of password and token, or any other suitable method. This would be limited by the network connectivity of the CS Platform to the remote host site and a back-up (or local) common CSE or user role password would need to be supplied.

The platform even can include a process for remotely resetting local passwords that are forgotten. The SA calls the help desk and is successfully authenticated as who they say they are. The Help desk places an order to the CS Platform (identified by IOT serial number) to have it reset its SA password. The SA is told to manually press the SYNC button, causing the CS Platform to check in with the Edge Host 410, receive the

order to reset the SA password, and complete the operation. If all else fails a CS Platform Factory Reset procedure can be followed which will reset all the passwords to default configuration.

New software services 140 can be added to the CS Platform add-on component through the normal DCS service subscription and activation processes. Subscribed services can be automatically managed and installed by the DMA 120 and the DCS end-to-end system 1. This enables the CS Platform 115 to offer new services over time. New software upgrades can be offered through the remote software upgrade feature of the CS Platform. This enables more significant upgrades of the CS Platform 115 to be performed with user approval without the need for a tech rep to visit the user's site. This increases the frequency at which system upgrades can be deployed because cost is significantly reduced.

New hardware can be added with the appropriate services added remotely to the platform because of the above features and the use of COTS technology for most of the system hardware and software. An example of a new service requiring hardware extensions would be web camera based support for users. With the addition of a low cost USB web cam, the CS Platform 115 can offer a service for those subscribed that allows them to get better remote support on the phone because they can snap and send photos of the problems they are having to a help desk or call center.

Embodiments contemplate installation of the CS Platform on a network connected personal computer on the same subnet as the CS platform 115. The install process, a schematic illustration of which is shown in FIG. 19, uses a combination of standard networking utilities and LED indications found on the back of the CS Platform to walk the installer through the process. Since the CS Platform 115 is preferably a headless embedded system, the install process can be tricky. The steps listed here are one possible way to do the install, though others are possible. The combination of feedback on the command screen and LEDs on the device provide a robust process for the installation. The component 115 is initially in power-on standby (block 801) and is

powered on by the user (block 802). Preferably, a status LED or the like blinks to indicate that the component 115 is booting, then becomes steady on when the component 115 is ready (block 803). In embodiments, the user reads the MAC address of the component 115 (block 804), opens a command window on the UI (block 805), and enters a command with the MAC address and other information (block 806). The user can then ping the component 115 (block 807) to test it, then wait for an indication of completion (block 808), such as one or more LEDs in a steady on state. The user then goes to the component's web server 130 via a browser (block 809), logs on as the administrator (block 810), and configures network information as required (block 811) to enable the component 115 to communicate with the edge host 410. The component 115 reboots, during which the IOT should be powered down (block 813). Once both have completed their reboot, installation and setup are complete (block 813).

The CS Platform can be configured in multiple ways for network connectivity, including use of a fixed IP address and use of DHCP to acquire an IP address. A fixed address is preferred for most users and has the advantage of making it easy to point a browser to the CS Platform UI when ready. DHCP is a very easy to install alternative, but would require a device domain name for the CS Platform and DNS services connection. One possible way to provide an automatic domain name is to combine the IOT serial number with the last two digits of the MAC address. Other combinations of readily available information known to the user and the CS Platform by default are possible.

The CS Platform is configured for the network just as any browser would be configured. This can be done manually via form fill-in on the CS Platform UI. It can also be done through a look-up to see the setting already found in the web browser platform if the OS provides that capability. This would provide the base-line settings and the user then has the ability to customize or correct them as necessary. Once configured, a Test Configuration button can be provided that immediately tries to contact the supplier edge server 410 to ensure that the settings are correct before the user leaves the network administration page.

Users of the CS Platform 115 may lose a bookmark to the CS Platform Web Page and a way needs to be provided for users to find that web page again easily. If DHCP was used to configure the system, then the user can simply follow instructions to determine the default or hard coded domain name of the CS Platform. A discovery tool can also be provided that is installed and runs on the DFE or a networked personal computer in the user's environment and will find and display all CS Platforms that are running. This discovery tool could also be downloaded from the supplier web site. A link to the tool could be made available from the CS Platform UI so the tool can be downloaded and saved in preparation for CS Platform IP address loss. The tool could also come stored locally on the CS Platform with an option to save it locally on the DFE during install.

As mentioned above, the router manages simultaneous access methods to the CS Platform. The CS Platform router is preferably compatible with the supplier gateway and DMA requirements and with devices 110 in the field. The router preferably provides direct connectivity between the Local PWS port and the IOT diagnostic (serial) port. The router of embodiments also provides network connectivity for a network client through, for example, the IOT diagnostic (serial) port, and can support network pathways to the IOT CAN Bus, to the EP Service for various devices 110, and mediates all (except EP) communications traffic and priorities. Priorities are enabled to allow the smooth transition of operational modes. For example, a priority for Application Session for DCU Software Upgrade, another for Local PWS Port IOT Diagnostic Session at the IOT Serial port3, and another for other Network Sessions. Under normal system operation, an "open" Local PWS session is preferably not pre-empted, and a Local PWS session request may preferably interrupt a network diagnostic application session. Any interruption should be graceful. Preemption of a Network DCU software upgrade session is preferably not be permissible, though an EP and/or CAN Bus session should preferably be permitted at any time as long as only one CAN session is permitted at any one time. The DMA EP Gateway server/client client preferably has exclusive access to the EP port,

and DCU V2.0 preferably supports a pathway for local PWS transactions over the network.

In support of DCU v2.0, the Communication Controller can provide, for example, the contents of local CSE diagnostic session over the network by generating a START\_DIAG\_SESSION event (Consumer of this event is DMA Push Event Provider), generating END\_DIAG\_SESSION event (Consumer of this event is DMA Push Event Provider), delivering each message between LOCAL\_PWS\_PORT and IOT\_SERIAL communication ports as PWS\_MESSAGE\_EVENT to DMA Push Event Provider.

In further support of DCU v2.0, the Communication Controller can support a local CSE diagnostic connection through the Local PWS port to the IOT serial port by providing a method to communicate its presence to the PWS connected to the system, for example by utilizing the RS232 signal designated as the CTS (Clear To Send) signal. The CTS signal is preferably held at logic level HIGH at the DCU.

The DSR (Data Signal Ready) signal in the RS232 interface can control the diagnostic mode of the IOT. The DCU will preserve or as necessary ‘create’ this control. If the IOT is in the diagnostic mode and is powered off it will power up in the diagnostics mode when the DSR signal to the IOT has been set HIGH. The PSW controls the DSR signal.

While particular embodiments have been described, alternatives, modifications, variations, improvements, and substantial equivalents that are or may be presently unforeseen may arise to applicants or others skilled in the art. Accordingly, the appended claims as filed and as they may be amended are intended to embrace all such alternatives, modifications variations, improvements, and substantial equivalents.

FIG. 1 is a schematic illustration of the overall architecture of embodiments.

FIG. 2 is an another schematic illustration of the overall architecture of embodiments.

FIG. 3 is a schematic illustration of a method of service subscription and deployment according to embodiments.

FIG. 4 is a schematic illustration of a deployment option according to embodiments.

FIG. 5 is a schematic illustration of an additional deployment option according to embodiments.

FIG. 6 is a schematic illustration of an additional deployment option according to embodiments.

FIG. 7 is a schematic illustration of an additional deployment option according to embodiments.

FIG. 8 is a schematic illustration of an additional deployment option according to embodiments.

FIG. 9 is a schematic illustration of an additional deployment option according to embodiments.

FIG. 10 is a more detailed schematic illustration of the device model agent according to embodiments more detailed schematic illustration of the interaction between devices, the device model agent, services proxies, and the services host according to embodiments.

FIG. 11 is a more detailed schematic illustration of the interaction between devices, the device model agent, services proxies, and the services host according to embodiments.

FIG. 12 is a schematic illustration of an additional deployment option according to embodiments.

FIG. 13 is a schematic illustration of an additional deployment option according to embodiments with more emphasis on the Device Model Agent and its relationship to a device.

FIG. 14 is a schematic illustration of an additional deployment option according to embodiments with more emphasis on the Device Model Agent and its relationship to a device from another perspective.

FIG. 15 is a schematic flow chart of a method of operation of a service manager of the Device Model Agent according to embodiments.

FIG. 16 is a schematic illustration of a CS Platform add-on component according to embodiments.

FIG. 17 is another schematic illustration of a CS Platform add-on component according to embodiments.

FIG. 18 is a schematic illustration of a wireless deployment scheme of a CS Platform add-on component according to embodiments.

FIG. 19 is a schematic illustration of a method of setting up a CS Platform add-on component according to embodiments.

FIG. 20 is a schematic illustration of a provisioning server according to embodiments.

FIG. 21 is a more schematic illustration of a CS Platform add-on component according to embodiments.

1. A distributed systems architecture comprising:

at least one device capable of providing at least one service available from a services host, the device including at least one device-specific provider application program interface and having device-specific status information;

at least one services layer; and

at least one device-independent runtime environment comprising:

at least one services environment in which the at least one service actually runs;

at least one common information management application program interface;

at least one device model agent; and

at least one common provider application program interface.

2. The architecture of claim 1 wherein the at least one services layer comprises at least one service made available to the at least one device.

3. The architecture of claim 1 wherein the at least one device-independent runtime environment is deployed in a marking device, the marking device being one of the at least one device.

4. The architecture of claim 1 wherein the at least one device-independent runtime environment is deployed in a server connected to a marking device, the marking device being one of the at least one device.

5. The architecture of claim 4 wherein the server hosts an application whose primary function is not related to the device-independent runtime environment, but which hosts the device-independent runtime environment.

6. The architecture of claim 1 wherein the at least one services environment and the services layer reside on a server connected to a marking device, the marking device being one of the at least one device, the server hosting an application whose primary function is not related to the services layer and environment, but which hosts them.

7. The architecture of claim 1 wherein the device model agent, the at least one services environment, and the services layer reside on a server connected to a marking device, the marking device being one of the at least one device, the server hosting an application whose primary function is not related to the device model agent and the services layer and environment, but which hosts them.

8. A method of providing device-independent services comprising:  
providing a common device interface;  
providing a common information model;  
integrating services in a device using the common device interface and information model; and  
hiding device-specific differences behind the common device interface.

9. The method of claim 8 wherein providing a common device interface comprises employing distributed model task force common information model with predetermined extensions for respective services.

10. The method of claim 9 wherein providing a common information model comprises basing the distributed model task force common information model with predetermined extensions enhancing compatibility between devices and respective services.

11. A method of providing a service platform comprising:  
providing an access module  
allowing services to use embedded computational power, data, and functions of a device via the access module; and  
deploying the access module in a common fashion.

12. The method of claim 11 further comprising accepting newly deployed services asynchronously with software releases for a hosting platform.

13. The method of claim 11 further comprising embedding the service platform in a host platform.

14. The method of claim 11 further comprising deploying the service platform in an add-on component to a host device.

15. The method of claim 14 further comprising connecting the add-on component to the host device via at least two interfaces.

16. The method of claim 14 further comprising connecting the add-on component to a network, thus providing the host device with the capability to participate in device services.

17. The method of claim 14 further comprising providing all network connectivity of the host device through the add-on component.

18. The method of claim 11 further comprising employing at least one application located in a user environment as a services proxy between at least one device and a services host.

19. The method of claim 18 wherein employing comprises sending data from the at least device to the services proxy in a first protocol and sending the data from the services proxy to the services host in a second protocol.

20. The method of claim 19 wherein the first protocol is SNMP.

21. The method of claim 19 wherein the first protocol is a wireless communications protocol.

22. The method of claim 19 further comprising managing device variations at a services host using a provisioning system for device-based services.

23. The method of claim 19 further comprising consolidating services management to a server in the user environment.

24. The method of claim 19 further comprising providing a UI with which a user can manage services.

25. A device model agent that provides an environment in which services can run substantially independent of a device for which the services are intended to provide functionality while providing access to the device, the device model agent also communicating with at least one services host to allow automated supplies maintenance and services subscription and deployment.

A distributed system (1) allows marking devices (110) and the like to subscribe to and run device-centric services (140). A device model agent (120) allows the devices (110) to interact with service hosts (310, 410) of service providers (300) to automate supplies maintenance, user help, and services subscription and deployment (501-506, 510-516, 900-908). The device model agent (120) can be embedded in the devices (110), can be deployed in an add-on component (115) connected to the device (110), or can be run by a separate machine as a proxy (220). The device model agent (120) provides a run time environment (124) for services (140) available to the device (110), but with a common interface (via 130) and a common structure so that services can be written once and run in the device model agent (120) in virtually any deployment.

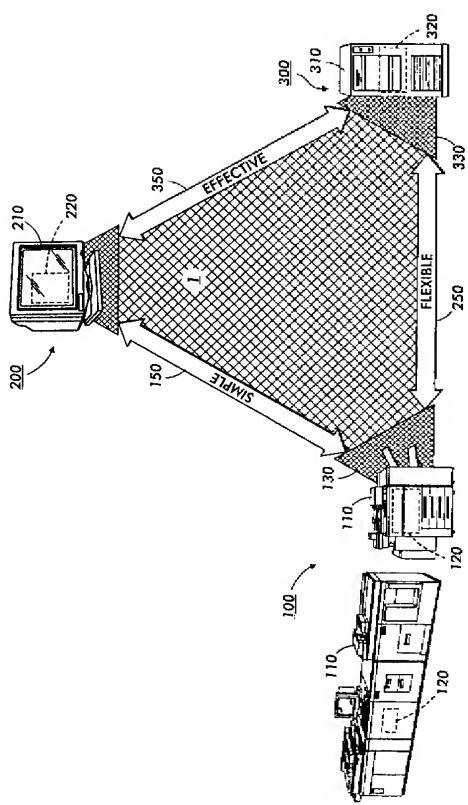


FIG. 1

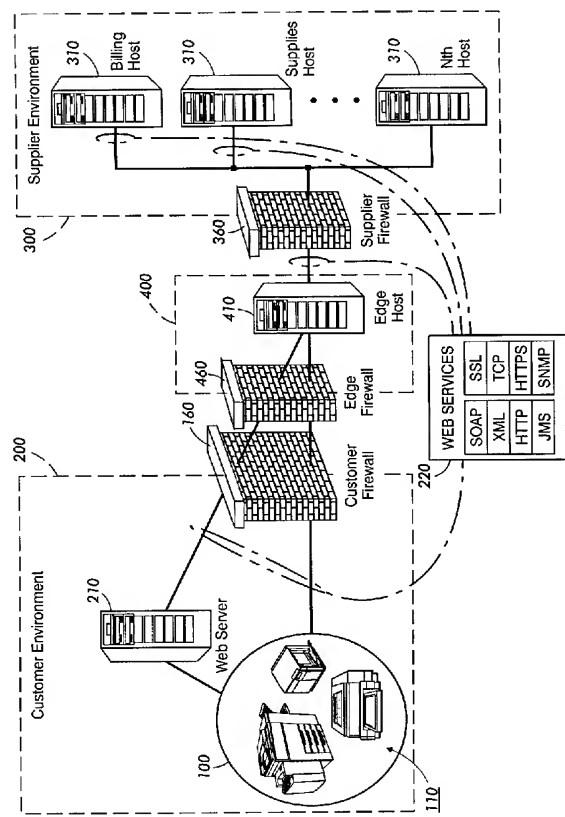


FIG. 2

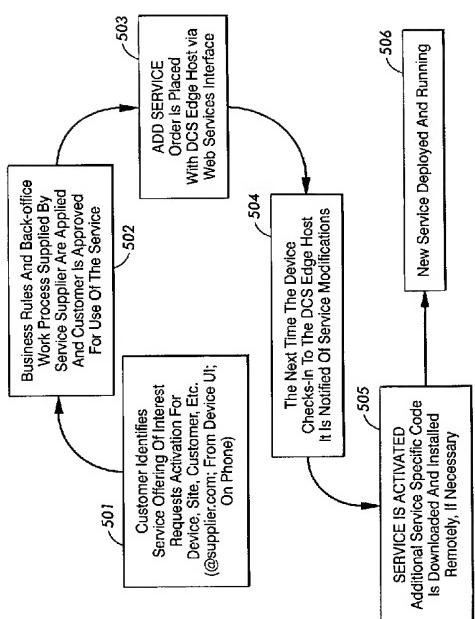


FIG. 3

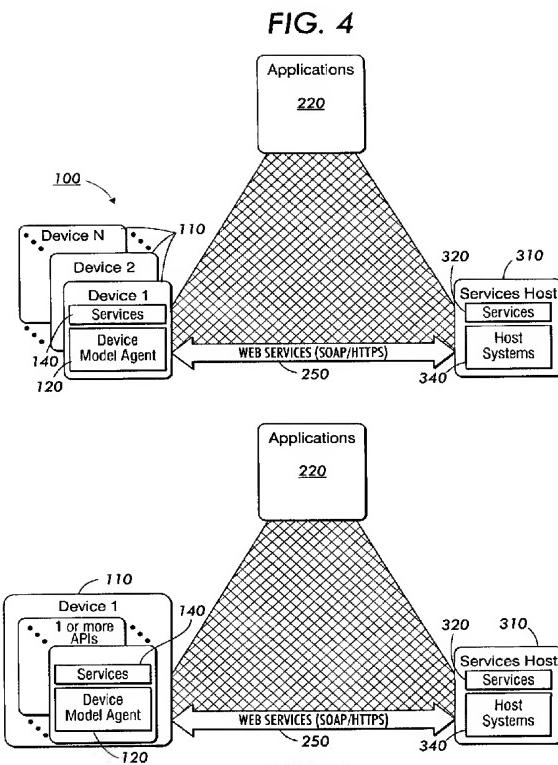


FIG. 5

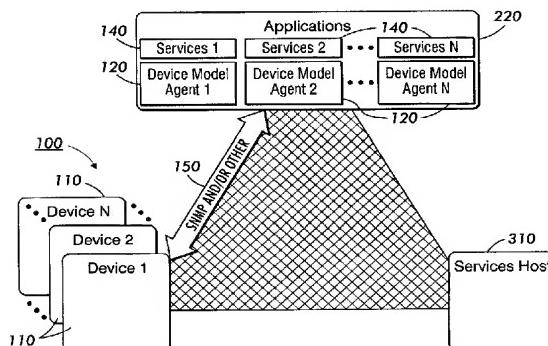
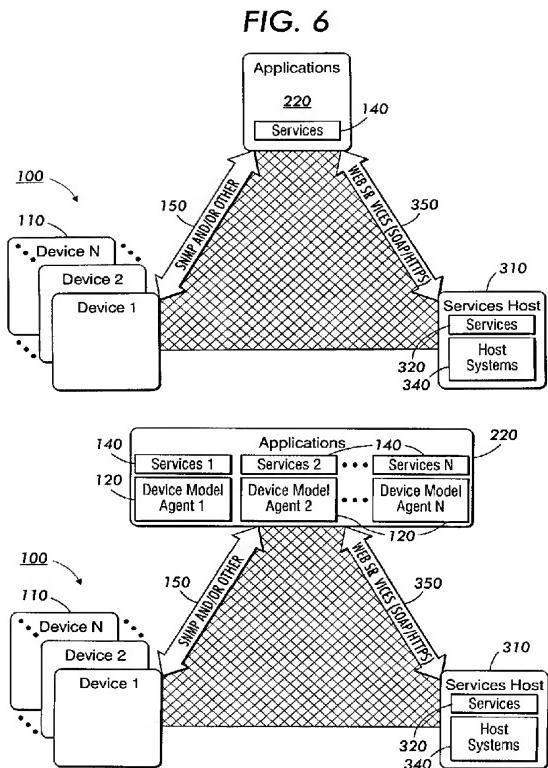


FIG. 8

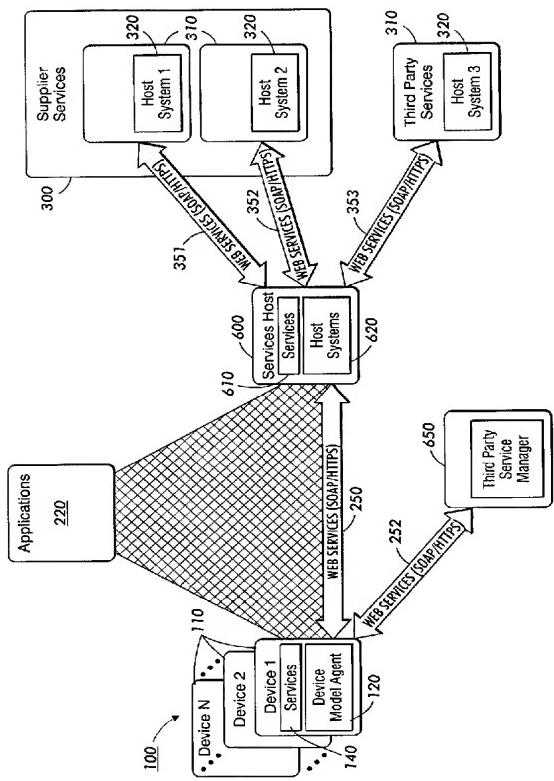


FIG. 9

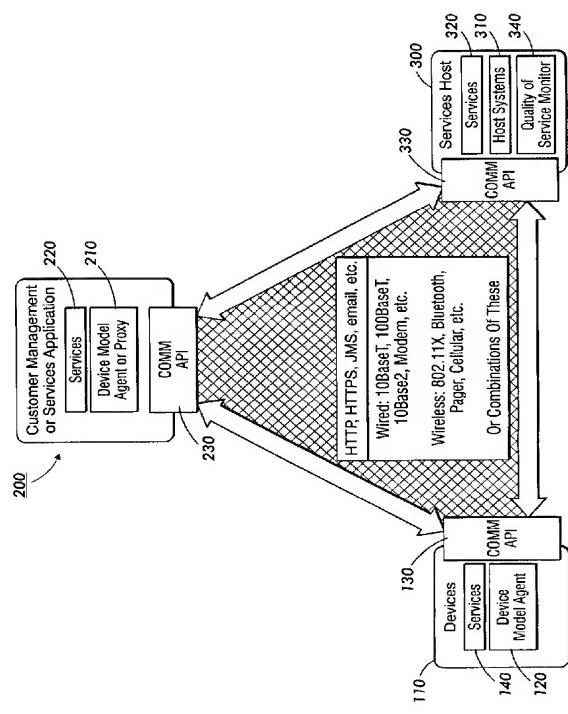


FIG. 11

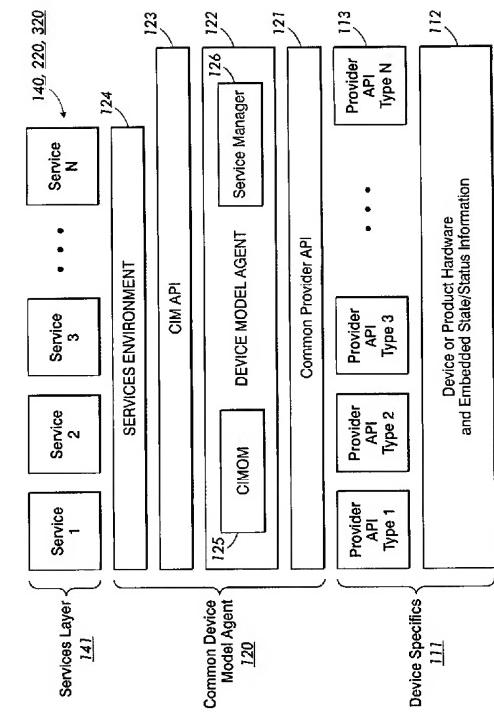


FIG. 10

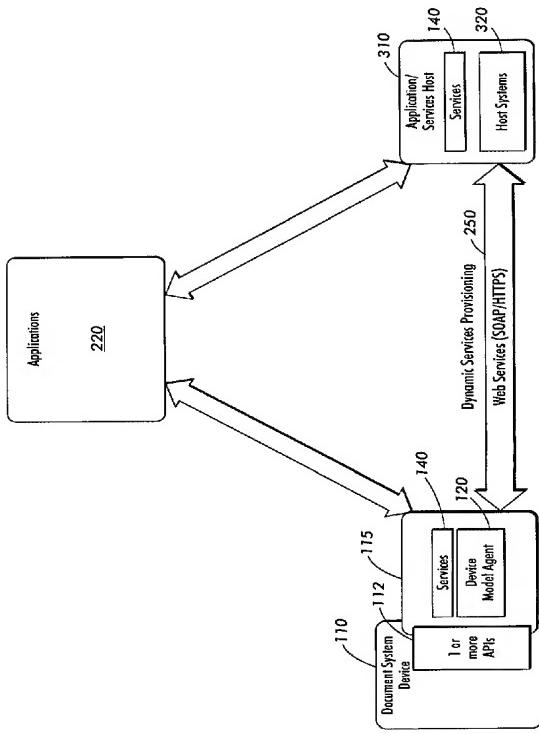


FIG. 12

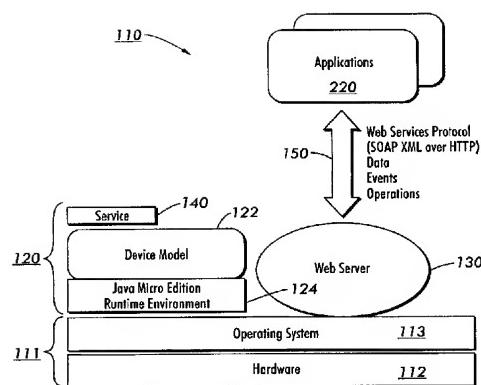


FIG. 13

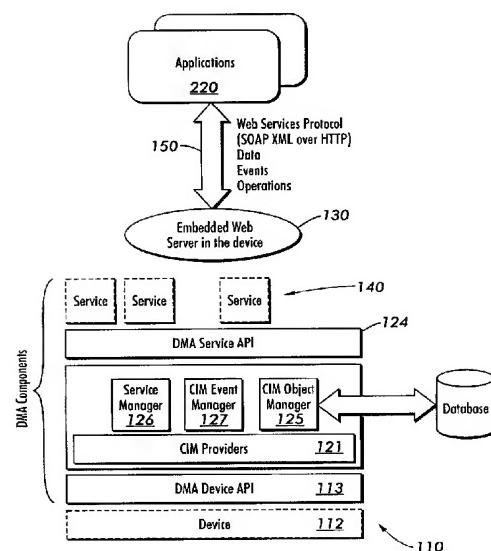


FIG. 14

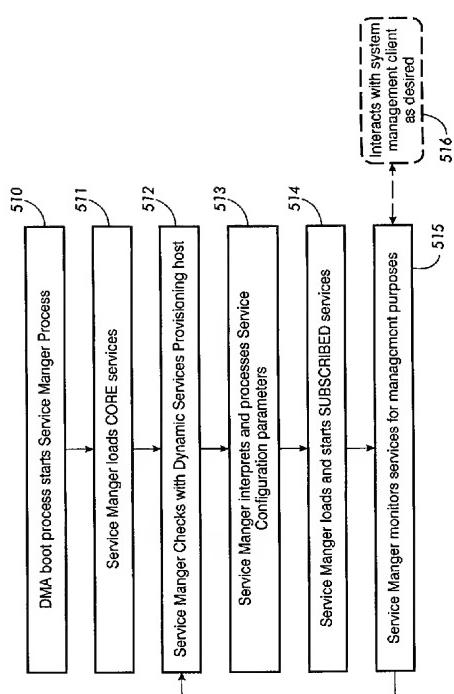


FIG. 15

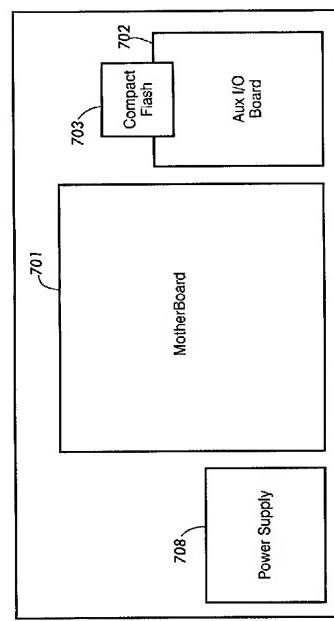


FIG. 16

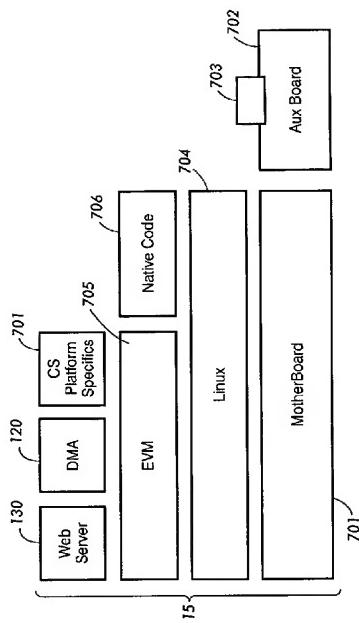


FIG. 17

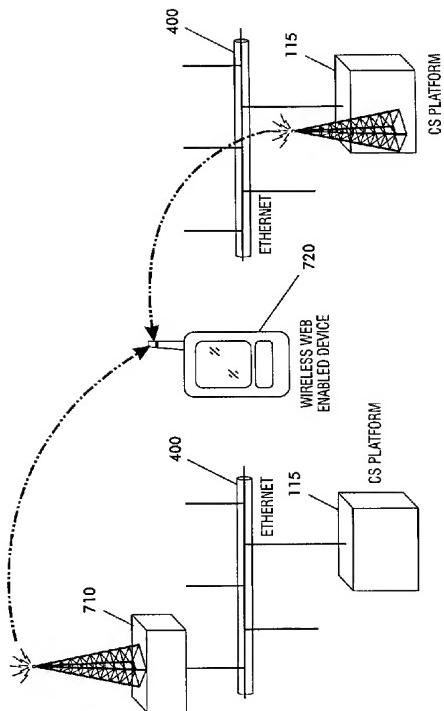


FIG. 18

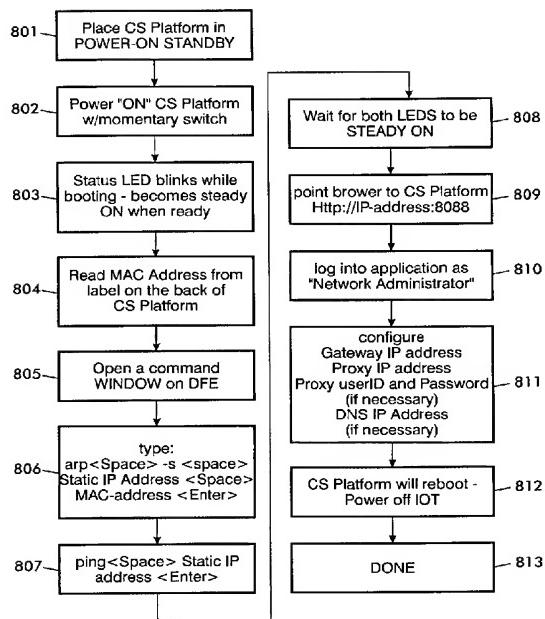


FIG. 19

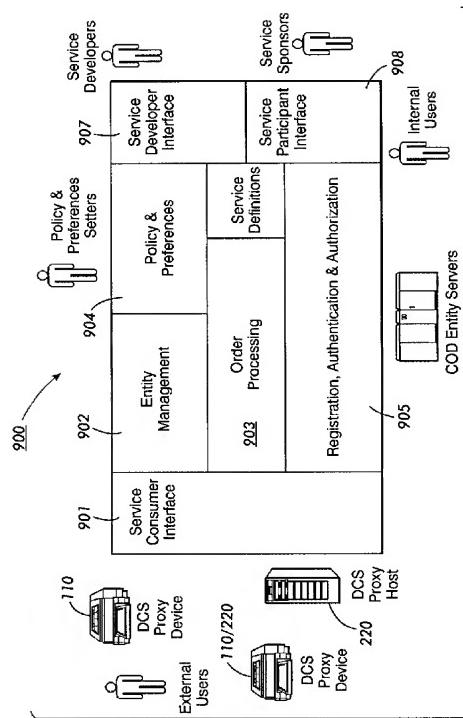


FIG. 20

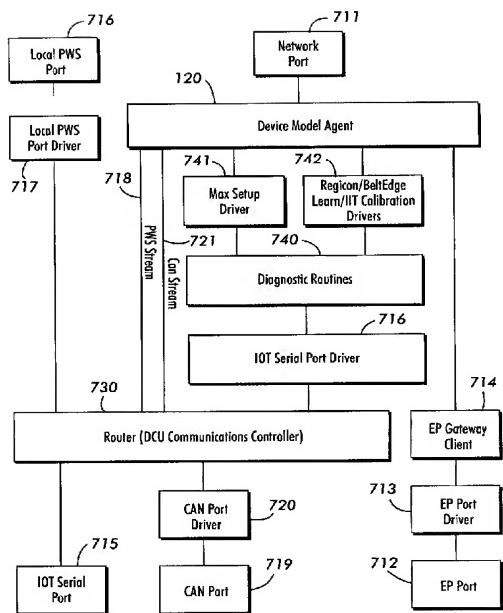


FIG. 21

**PAT-NO:** JP02004220556A  
**DOCUMENT-IDENTIFIER:** JP 2004220556 A  
**TITLE:** DISTRIBUTED SUBSCRIPTION SERVICE INCLUDING DEVICE MODEL AGENT AND ADD-ON COMPONENT, METHOD AND DEVICE ENABLING AUTOMATIC SUPPLY MAINTENANCE AND DEVICE INDEPENDENT SERVICE, LOW-COST BUILT-IN PLATFORM IN DEVICE SIDE AND METHOD AND DEVICE FOR INTEGRATED SERVER PLATFORM FOR EXECUTING DISTRIBUTED SERVICE AND SPONTANEOUS PROVISION OF DEVICE SERVICE  
**PUBN-DATE:** August 5, 2004

**INVENTOR-INFORMATION:**

<b>NAME</b>	<b>COUNTRY</b>
MICHAEL, R FAST	N/A
ROCKWELL, RONALD M	N/A
SHARMA, NAVEEN	N/A
FILLION, CLAUDE S	N/A
KEHOE, MICHAEL P	N/A
LORENZO, ARTURO M	N/A
MCCORKINDALE, MARY C	N/A
ST, JACQUES ROBERT J	N/A
THIERET, TRACY E	N/A
AUSTIN, JOHN C	N/A
DANIELS, MARC D	N/A
CAVANAUGH, MICHAEL F	N/A

HUANG, WEIXIA	N/A
REGRUIT, CHRISTOPHER J	N/A
WHITFIELD, LORANZO	N/A

**ASSIGNEE-INFORMATION:**

<b>NAME</b>	<b>COUNTRY</b>
XEROX CORP	N/A

**APPL-NO:** JP2003358032

**APPL-DATE:** October 17, 2003

**PRIORITY-DATA:** 2002319623 (October 17, 2002) ,  
2002319624 (October 17, 2002) ,  
2002319625 (October 17, 2002)

**INT-CL (IPC):** G06F009/445 , H04N001/00

**ABSTRACT:**

**PROBLEM TO BE SOLVED:** To provide a method flexibly supplying improvement of a device after the delivery to a user.

**SOLUTION:** A distributed system 1 allows a recording device 110 and the like to subscribe to and run device-centric services 140. A device model agent 120 allows the device 110 to interact with service host 310 of a service provider 300 to automate supplies maintenance, user help and services subscription. The device model agent 120 can be embedded in the device 110, can be deployed in an add-on component 115 connected to the device

110 or executed by separate machine as a proxy  
220. The device model agent 120 provides a run  
time environment 121 for services 140 available to  
the device 110 via a common interface along with a  
common structure.

COPYRIGHT: (C) 2004, JPO&NCIPI